Softwares Aiding with Data Recovery

INFR 4690U: I.T Forensics
April 5th, 2020
Group #9

Prepared by:

Bhavik Naik - 100617351
Imad Khan – 100621869
Jason Pinto - 100592099

Table of Contents

1 - Scope of Work 1

2 - TestDisk 1

2.1 - What is TestDisk? 1

2.2 - My Opinion about TestDisk 1

2.3 - How does it work? 2

2.4 - Live Demo: TestDisk 2

3 - The Sleuth Kit (TSK) 10

3.1 - What is The Sleuth Kit 10

3.2 - My Opinion about The Sleuth Kit 10

3.3 - How does it work? 10

3.4 - Live Demo: The Sleuth Kit 11

4 - TestDisk vs. The Sleuth Kit (TSK) 15

5 - AccessData Forensic Toolkit 15

6 - Autopsy 19

7 - Foremost 24

8 - ntfsundelete 27

9 - Challenge with ntfsundelete 30

10 - References 33

1 - Scope of Work

Our team must demonstrate how deleted files on a USB can be retrieved. Giving knowledge about deleted files on a USB are not permanently deleted from memory is important to learn. Introducing these skills and tools are effective in a wide variety of situations such as: extracting evidence of a criminal activity from a deleted file or retrieving files that were accidentally deleted. We will showcase six different softwares for data recovery and their advantages and disadvantages.

To find more information about this report, we have done a blog post that can be found here: https://bhaviknaik.medium.com/softwares-aiding-with-data-recovery-3977bc3a3a46

To see the installation and live demo tutorial, please see our video: https://drive.google.com/file/d/1KD_AjLX7McWJPUyUoHHhIOdeN6bLjBk3/view?usp=sharing

2 - TestDisk

2.1 - What is TestDisk?

TestDisk is a useful and powerful open-source data recovery software [1] which was created in 1998 [2]. This is a useful software as it can recover data from incidents caused by faulty software, certain types of viruses and/or human error [1]. TestDisk is used for many different data recovery situations which include:

- Fixed and/or deleted partition table [1]
- Recover FAT32 boot sector from its backup [1]
- Rebuild all FAT boot sector (FAT12/FAT16/FAT32) [1]
- Rebuild NTFS boot sector [1]
- Recover NTFS boot sector from its backup [1]
- Fix MFT using MFT mirror [1]
- Locate ext2/ext3/ext4 backup superblock [1]
- Undelete files from FAT, exFAT, NTFS and ext2 filesystem [1]
- Copy files from deleted FAT, exFAT, NTFS and ext2/ext3/ext4 partitions [1]

2.2 - My Opinion about TestDisk

TestDisk in my opinion is amazing because both novices and experts [1] in forensics are able to utilize it while having a decent variety of options to choose from in regard to data recovery. In addition, TestDisk is not a proprietary software on one given operating system. It works on multiple different operating systems which include:

- DOS (either real or in a Windows 9x DOS-box) [1]
- Windows 10/8.1/8/7/Vista/XP, Windows Server 2016/2012/2008/2003 [1]
- Linux [1]
- FreeBSD, NetBSD, OpenBSD [1]
- SunOS [1]
- MacOS [1]

2.3 - How does it work?

It uses its own libraries and commands in order to extract different data depending on the user. For our example we will recover a .docx file. TestDisk recovers many different file formats which includes:

- JPEG
- Pdf
- PNG
- Doc
- txt

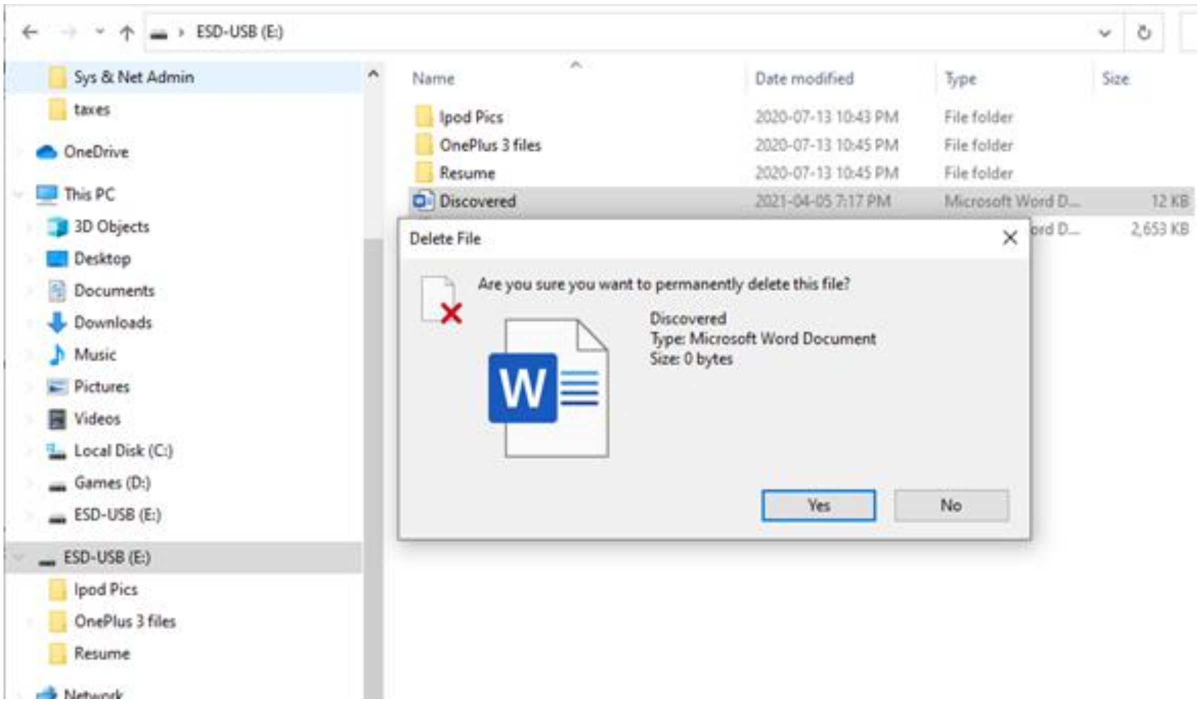
We will go over functionality and a live demo in more details (see live demo: TestDisk). For now, we will go over the general details of TestDisk.

1. Once TestDisk is booted up, it will detect hard drives automatically followed by the correct size in each hard drive [2]
2. Select the hard drive that you would like to recover or rebuild data and/or files with your arrow keys on the keyboard and once hovered on the right hard drive, press “enter”
3. TestDisk will auto-detect the partition table. From there we can select the partition table to enter by once again using our arrow keys and pressing “enter”
4. Once you enter a particular table you will enter “advanced” option and select the partition that has the deleted file and choose “undelete” [2]
5. After this, TestDisk will scan MFT entries for deleted files and showcase a list of NTFSs deleted files
6. Select the file you wish to recover and select the destination in your hard drive where you would like to store the file.

TestDisk makes it easy and efficient to retrieve deleted files. Users with novice IT skills can utilize TestDisk which is why it is so powerful. It has many powerful features while being easy to use.

2.4 - Live Demo: TestDisk

For this live demo we will be extracting a .docx file to showcase how to recover the deleted file from TestDisk.



Name	Date modified	Type	Size
Ipod Pics	2020-07-13 10:43 PM	File folder	
OnePlus 3 files	2020-07-13 10:45 PM	File folder	
Resume	2020-07-13 10:45 PM	File folder	
Discovered	2021-04-05 7:17 PM	Microsoft Word D...	12 KB
		ord D...	2,653 KB

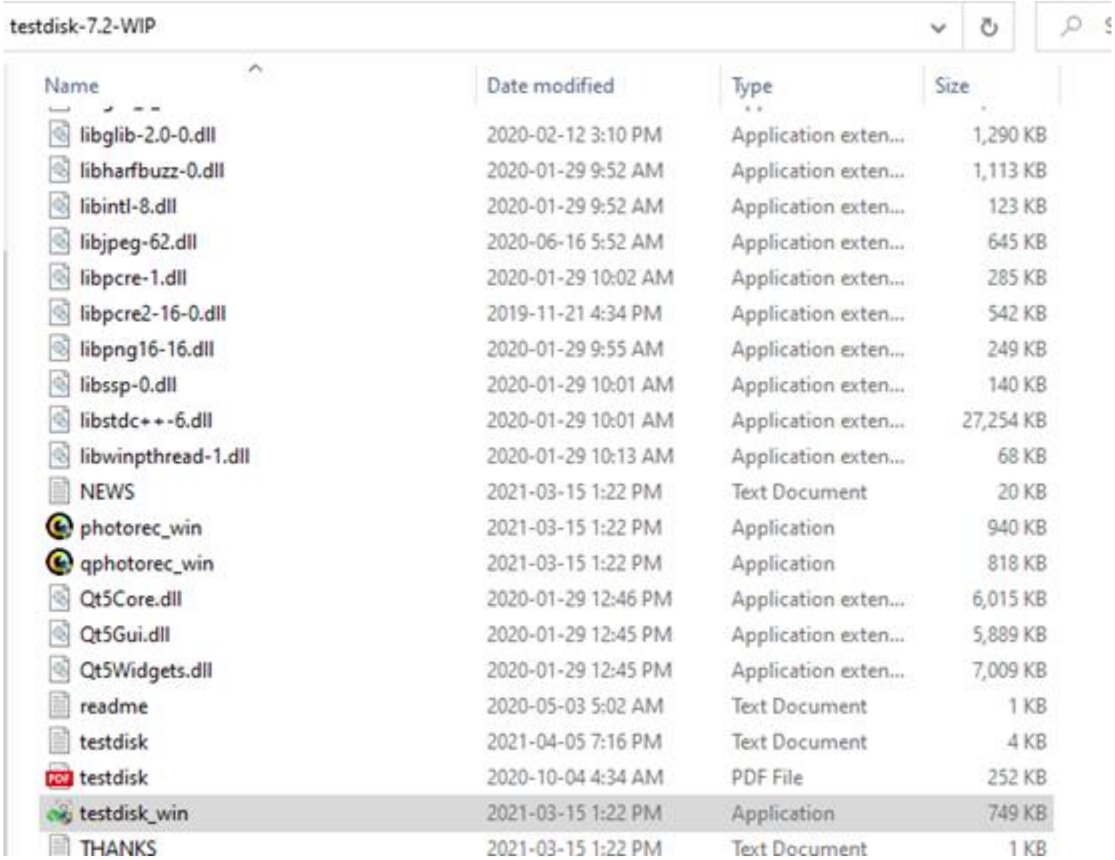
Delete File

Are you sure you want to permanently delete this file?

Discovered
Type: Microsoft Word Document
Size: 0 bytes

Figure 1.1: Deleting the .doc file form the USB.

The first step in our recovering process is deleting a file to showcase how to recover it. We will delete a file called “Discovered.docx” and recover it. It is important to note that although a message pops up indicating that the file will be permanently deleted (*see figure 1.1*), this is false, and the metadata of the file is still stored in the USB.

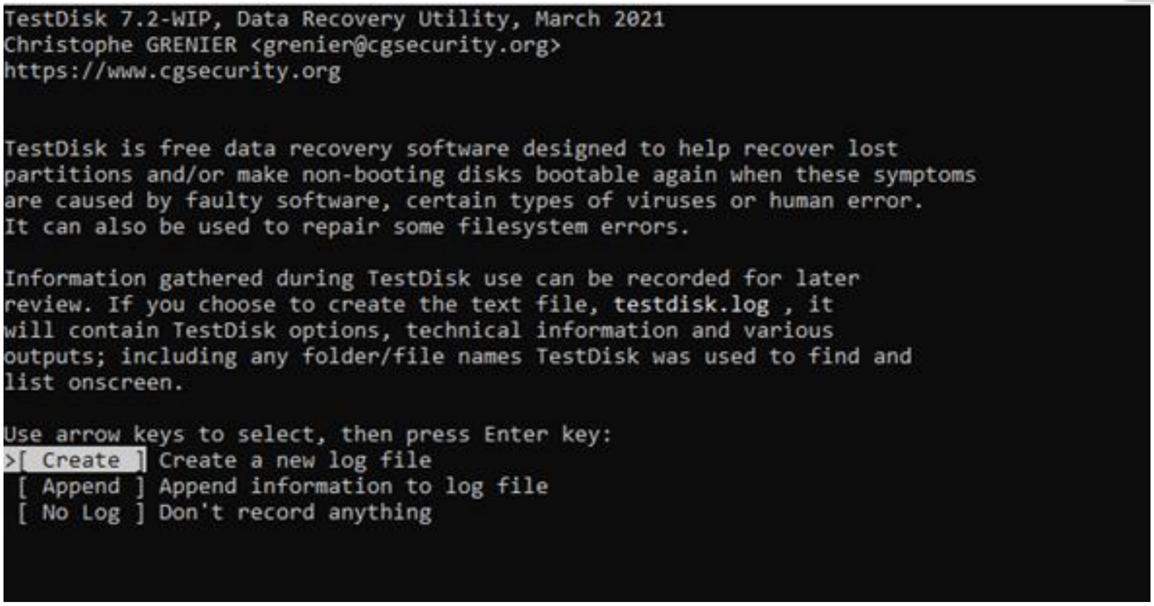


testdisk-7.2-WIP

Name	Date modified	Type	Size
libglib-2.0-0.dll	2020-02-12 3:10 PM	Application exten...	1,290 KB
libharfbuzz-0.dll	2020-01-29 9:52 AM	Application exten...	1,113 KB
libintl-8.dll	2020-01-29 9:52 AM	Application exten...	123 KB
libjpeg-62.dll	2020-06-16 5:52 AM	Application exten...	645 KB
libpcre-1.dll	2020-01-29 10:02 AM	Application exten...	285 KB
libpcre2-16-0.dll	2019-11-21 4:34 PM	Application exten...	542 KB
libpng16-16.dll	2020-01-29 9:55 AM	Application exten...	249 KB
libssp-0.dll	2020-01-29 10:01 AM	Application exten...	140 KB
libstdc++-6.dll	2020-01-29 10:01 AM	Application exten...	27,254 KB
libwinpthread-1.dll	2020-01-29 10:13 AM	Application exten...	68 KB
NEWS	2021-03-15 1:22 PM	Text Document	20 KB
photorec_win	2021-03-15 1:22 PM	Application	940 KB
qphotorec_win	2021-03-15 1:22 PM	Application	818 KB
Qt5Core.dll	2020-01-29 12:46 PM	Application exten...	6,015 KB
Qt5Gui.dll	2020-01-29 12:45 PM	Application exten...	5,889 KB
Qt5Widgets.dll	2020-01-29 12:45 PM	Application exten...	7,009 KB
readme	2020-05-03 5:02 AM	Text Document	1 KB
testdisk	2021-04-05 7:16 PM	Text Document	4 KB
testdisk	2020-10-04 4:34 AM	PDF File	252 KB
testdisk_win	2021-03-15 1:22 PM	Application	749 KB
THANKS	2021-03-15 1:22 PM	Text Document	1 KB

Figure 1.2: The extracted TestDisk zip file and is content

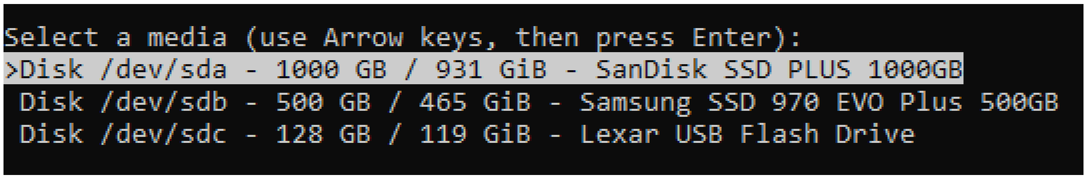
Now we open the application of TestDisk in order to run the software. The highlighted application in *figure 1.2* is the application we wish to open.



```
TestDisk 7.2-WIP, Data Recovery Utility, March 2021
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org


TestDisk is free data recovery software designed to help recover lost
partitions and/or make non-booting disks bootable again when these symptoms
are caused by faulty software, certain types of viruses or human error.
It can also be used to repair some filesystem errors.

Information gathered during TestDisk use can be recorded for later
review. If you choose to create the text file, testdisk.log , it
will contain TestDisk options, technical information and various
outputs; including any folder/file names TestDisk was used to find and
list onscreen.

Use arrow keys to select, then press Enter key:
>[ Create ] Create a new log file
 [ Append ] Append information to log file
 [ No Log ] Don't record anything
```

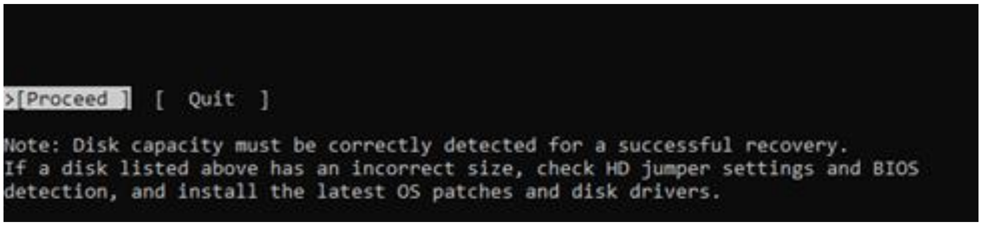
Figure 1.3: The startup options for TestDisk

In TestDisk you get to choose from 3 options. You should usually pick “create” unless there is a reason to append data to the log or if you execute TestDisk read only media and cannot create it anywhere else [2].



```
Select a media (use Arrow keys, then press Enter):
>Disk /dev/sda - 1000 GB / 931 GiB - SanDisk SSD PLUS 1000GB
 Disk /dev/sdb - 500 GB / 465 GiB - Samsung SSD 970 EVO Plus 500GB
 Disk /dev/sdc - 128 GB / 119 GiB - Lexar USB Flash Drive
```

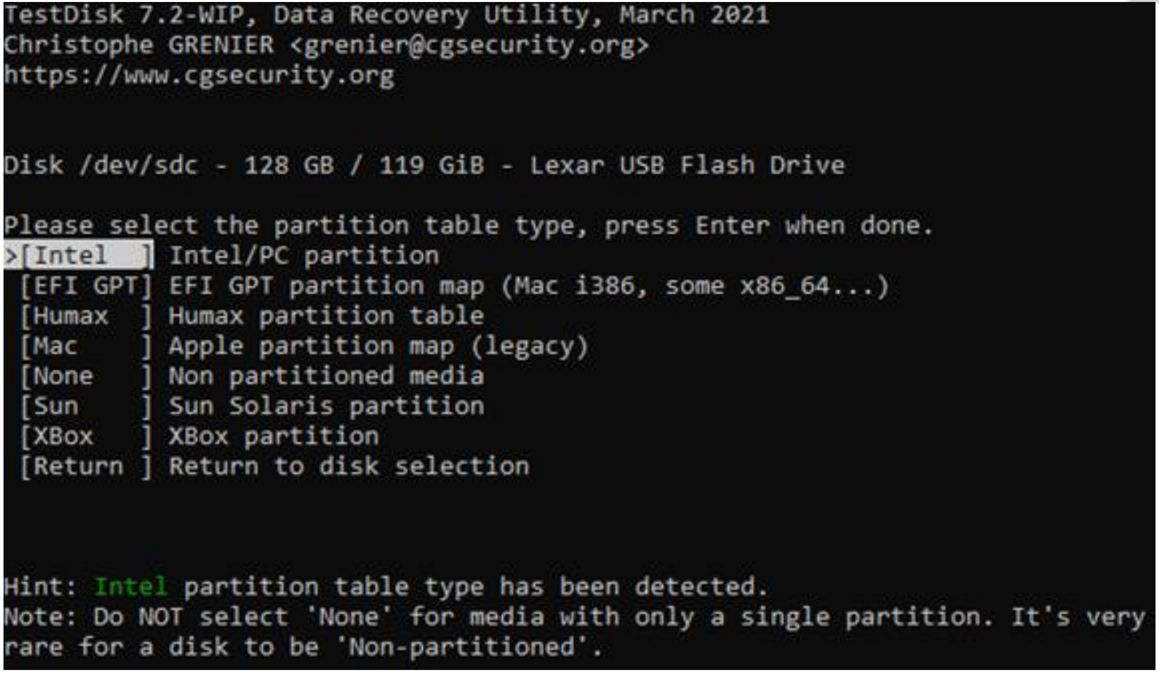
Figure 1.4: All the hard drives on the local computer



```
>[Proceed ]  [  Quit  ]

Note: Disk capacity must be correctly detected for a successful recovery.
If a disk listed above has an incorrect size, check HD jumper settings and BIOS
detection, and install the latest OS patches and disk drivers.
```

Figure 1.5: The options we can choose from

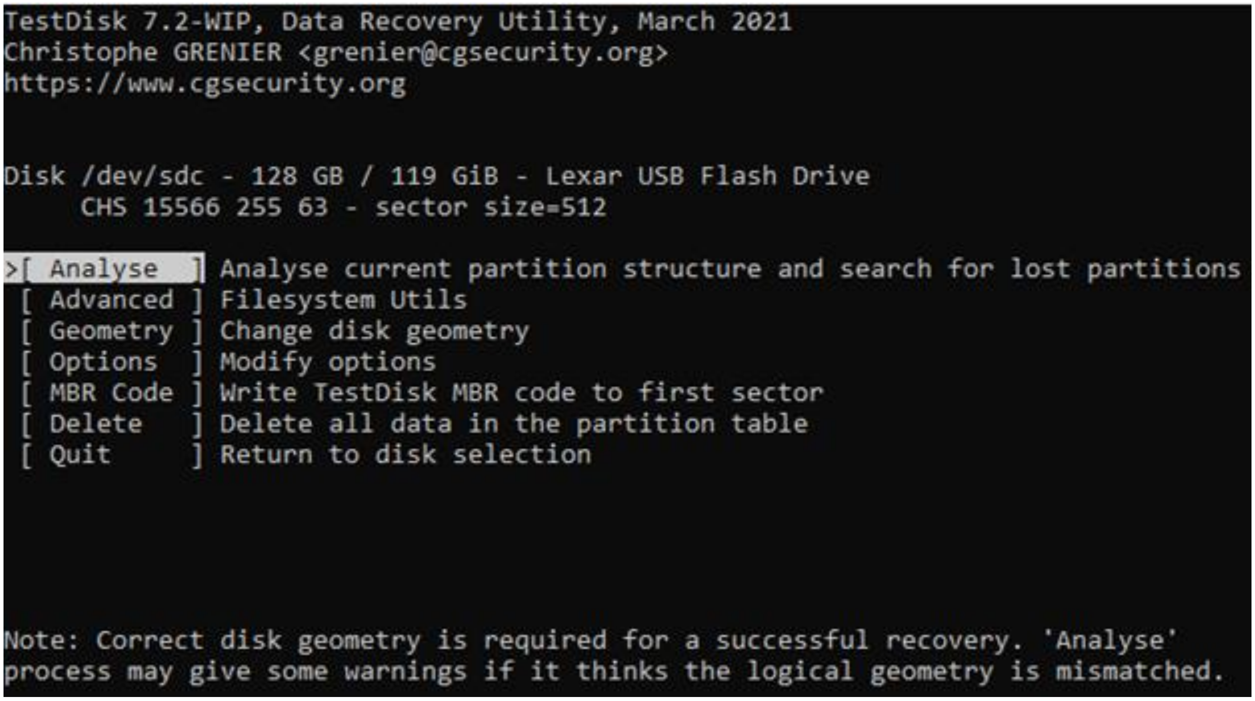
Once you click one of the three options, you will be greeted with the hard drives and devices on your local computer. For simplicity’s sake we will recover the file from the option “create.” All hard drives are automatically detected and listed by TestDisk. TestDisk also shows the size of the hard rives and the description of them which is very convenient (see Figure 1.4). Since we want to recover a deleted file from our USB, we will pick the third option as it is conveniently labelled “Lexar USB flash drive” (see figure 1.4). At the bottom of the page, we will see two option for “Proceed” or “Quit” (see figure 1.5) which we can use to continue with our recovery.



```
TestDisk 7.2-WIP, Data Recovery Utility, March 2021
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org


Disk /dev/sdc - 128 GB / 119 GiB - Lexar USB Flash Drive

Please select the partition table type, press Enter when done.
>[Intel   ] Intel/PC partition
 [EFI GPT] EFI GPT partition map (Mac i386, some x86_64...)
 [Humax   ] Humax partition table
 [Mac     ] Apple partition map (legacy)
 [None    ] Non partitioned media
 [Sun     ] Sun Solaris partition
 [XBox    ] XBox partition
 [Return  ] Return to disk selection



Hint: Intel partition table type has been detected.
Note: Do NOT select 'None' for media with only a single partition. It's very
rare for a disk to be 'Non-partitioned'.
```

Figure 1.6: all the partition tables in the USB

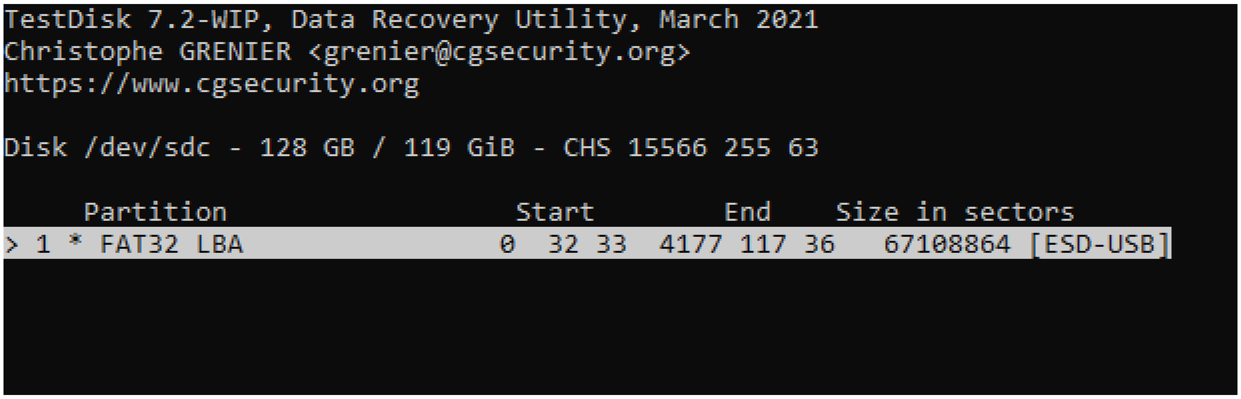
These are where our partition tables are found. We need to get into the right partition table to see where the file has been deleted. Most of these partition tables will not have any file data. Most of the files and folders are kept in the partition tables that TestDisk will automatically detect. In our example in figure 1.6, we can see that "Intel" is the partition table that TestDisk has found and so we will enter that partition table.



```
TestDisk 7.2-WIP, Data Recovery Utility, March 2021
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org


Disk /dev/sdc - 128 GB / 119 GiB - Lexar USB Flash Drive
     CHS 15566 255 63 - sector size=512

>[ Analyse  ] Analyse current partition structure and search for lost partitions
 [ Advanced ] Filesystem Utils
 [ Geometry ] Change disk geometry
 [ Options  ] Modify options
 [ MBR Code ] Write TestDisk MBR code to first sector
 [ Delete   ] Delete all data in the partition table
 [ Quit     ] Return to disk selection




Note: Correct disk geometry is required for a successful recovery. 'Analyse'
process may give some warnings if it thinks the logical geometry is mismatched.
```

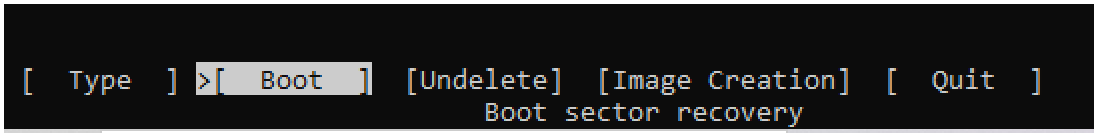
Figure 1.7: Options we can do inside the partition table.

Once we get into the partition table, there will be many options for us to choose from. Since we are trying to recover a deleted file, we must enter the “Advanced” option (*see figure 1.7*).



```
TestDisk 7.2-WIP, Data Recovery Utility, March 2021
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org

Disk /dev/sdc - 128 GB / 119 GiB - CHS 15566 255 63

     Partition                  Start        End    Size in sectors
> 1 * FAT32 LBA                0  32 33  4177 117 36   67108864 [ESD-USB]
```

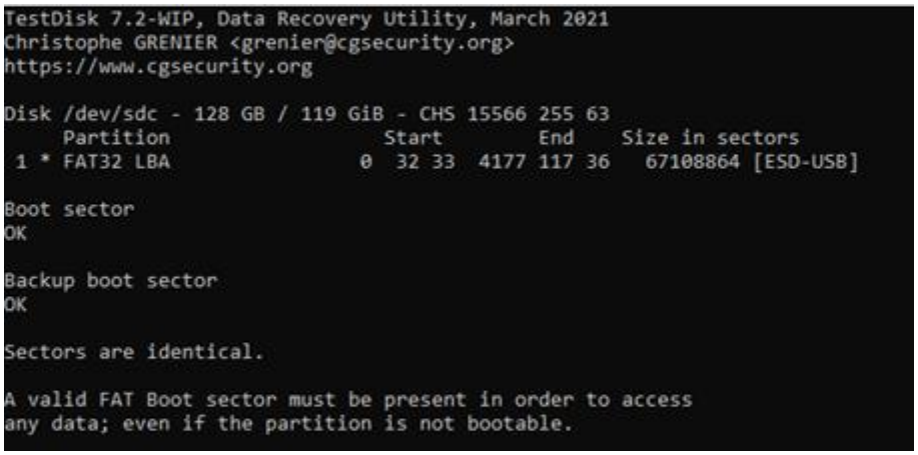
Figure 1.8: The partitions in the hard drive



```
[  Type  ] >[  Boot  ]  [Undelete]  [Image Creation]  [  Quit  ]
                             Boot sector recovery
```

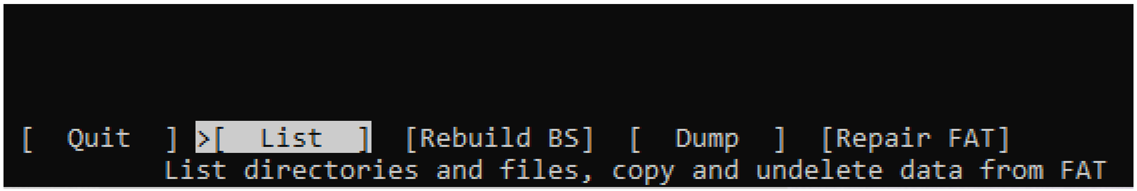
Figure 1.9: The different options we can select we can do to the partitions.

Once in Advanced, this is where our file system partitions will be listed. Since this USB only has one partition, it makes it easy where the files and folders are stored (*see figure 1.8*). However, it is not uncommon for USBs and hard drives to have more than one file system such as FAT32 and NTFS. We can do a couple of different options with the partition however since we are trying to access a deleted file in this partition we want to “boot” this partition (*see figure 1.9*).



```
TestDisk 7.2-WIP, Data Recovery Utility, March 2021
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org

Disk /dev/sdc - 128 GB / 119 GiB - CHS 15566 255 63
     Partition                  Start        End    Size in sectors
 1 * FAT32 LBA                0  32 33  4177 117 36   67108864 [ESD-USB]

Boot sector
OK

Backup boot sector
OK

Sectors are identical.

A valid FAT Boot sector must be present in order to access
any data; even if the partition is not bootable.
```

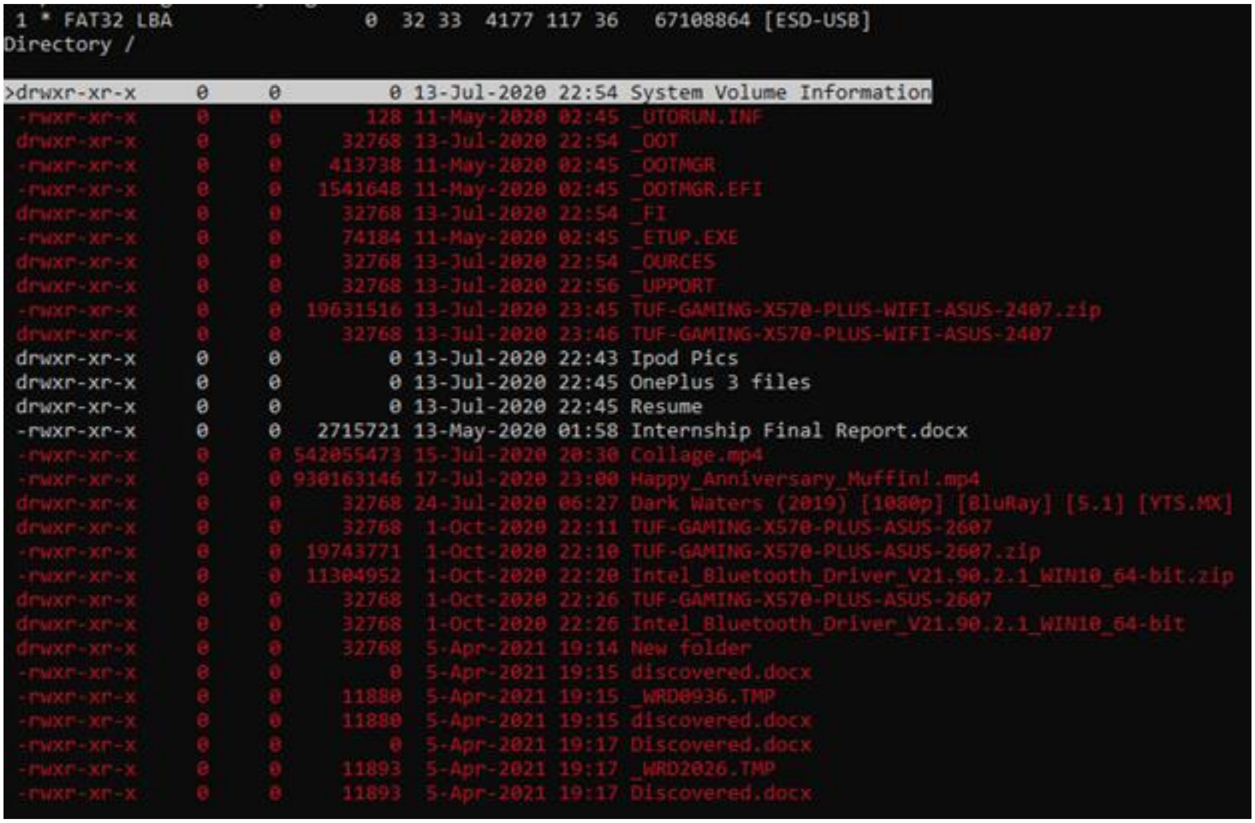
Figure 1.10: TestDisk confirms the partition and boot sector.



```
[  Quit  ] >[  List  ]  [Rebuild BS]  [  Dump  ]  [Repair FAT]
         List directories and files, copy and undelete data from FAT
```

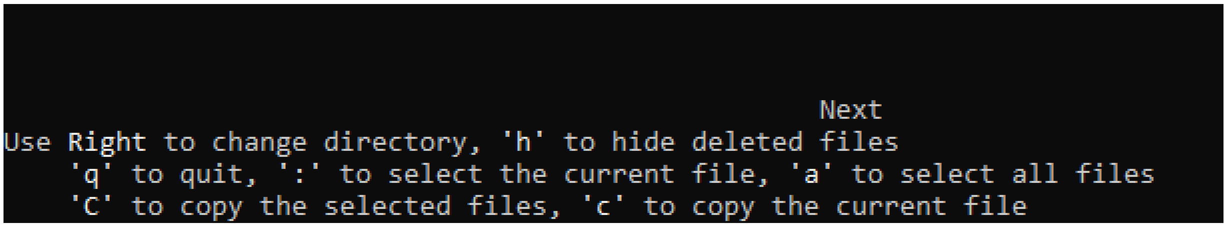
Figure 1.11: The options listed after TestDisk analyzes the specific partition

Once TestDisk analyzes the partition and confirms (see figure 1.10) it is okay to enter. We can use the options listed at the bottom (see figure 1.11) to hover over “List” and press enter.



```
 1 * FAT32 LBA              0  32 33  4177 117 36   67108864 [ESD-USB]
Directory /

>drwxr-xr-x     0     0         0 13-Jul-2020 22:54 System Volume Information
 -rwxr-xr-x     0     0       128 11-May-2020 02:45 _UTORUN.INF
 drwxr-xr-x     0     0     32768 13-Jul-2020 22:54 _OOT
 -rwxr-xr-x     0     0    413738 11-May-2020 02:45 _OOTMGR
 -rwxr-xr-x     0     0   1541648 11-May-2020 02:45 _OOTMGR.EFI
 drwxr-xr-x     0     0     32768 13-Jul-2020 22:54 _FI
 -rwxr-xr-x     0     0     74184 11-May-2020 02:45 _ETUP.EXE
 drwxr-xr-x     0     0     32768 13-Jul-2020 22:54 _OURCES
 drwxr-xr-x     0     0     32768 13-Jul-2020 22:56 _UPPORT
 -rwxr-xr-x     0     0  19631516 13-Jul-2020 23:45 TUF-GAMING-X570-PLUS-WIFI-ASUS-2407.zip
 drwxr-xr-x     0     0     32768 13-Jul-2020 23:46 TUF-GAMING-X570-PLUS-WIFI-ASUS-2407
 drwxr-xr-x     0     0         0 13-Jul-2020 22:43 Ipod Pics
 drwxr-xr-x     0     0         0 13-Jul-2020 22:45 OnePlus 3 files
 drwxr-xr-x     0     0         0 13-Jul-2020 22:45 Resume
 -rwxr-xr-x     0     0   2715721 13-May-2020 01:58 Internship Final Report.docx
 -rwxr-xr-x     0     0 542055473 15-Jul-2020 20:30 Collage.mp4
 -rwxr-xr-x     0     0 930163146 17-Jul-2020 23:00 Happy_Anniversary_Muffin!.mp4
 drwxr-xr-x     0     0     32768 24-Jul-2020 06:27 Dark Waters (2019) [1080p] [BluRay] [5.1] [YTS.MX]
 drwxr-xr-x     0     0     32768  1-Oct-2020 22:11 TUF-GAMING-X570-PLUS-ASUS-2607
 -rwxr-xr-x     0     0  19743771  1-Oct-2020 22:10 TUF-GAMING-X570-PLUS-ASUS-2607.zip
 -rwxr-xr-x     0     0  11304952  1-Oct-2020 22:20 Intel_Bluetooth_Driver_V21.90.2.1_WIN10_64-bit.zip
 drwxr-xr-x     0     0     32768  1-Oct-2020 22:26 TUF-GAMING-X570-PLUS-ASUS-2607
 drwxr-xr-x     0     0     32768  1-Oct-2020 22:26 Intel_Bluetooth_Driver_V21.90.2.1_WIN10_64-bit
 drwxr-xr-x     0     0     32768  5-Apr-2021 19:14 New folder
 -rwxr-xr-x     0     0         0  5-Apr-2021 19:15 discovered.docx
 -rwxr-xr-x     0     0     11880  5-Apr-2021 19:15 _WRD0936.TMP
 -rwxr-xr-x     0     0     11880  5-Apr-2021 19:15 discovered.docx
 -rwxr-xr-x     0     0         0  5-Apr-2021 19:17 Discovered.docx
 -rwxr-xr-x     0     0     11893  5-Apr-2021 19:17 _WRD2026.TMP
 -rwxr-xr-x     0     0     11893  5-Apr-2021 19:17 Discovered.docx
```

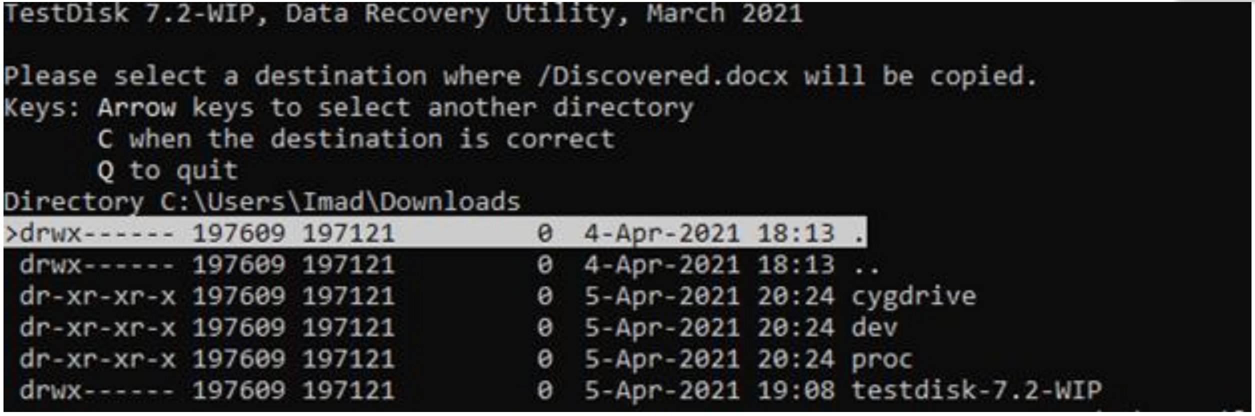
Figure 1.12: All files and folders that are active and deleted in the USB.



```
                                                   Next
Use Right to change directory, 'h' to hide deleted files
    'q' to quit, ':' to select the current file, 'a' to select all files
    'C' to copy the selected files, 'c' to copy the current file
```

Figure 13: The different types of options we can choose.

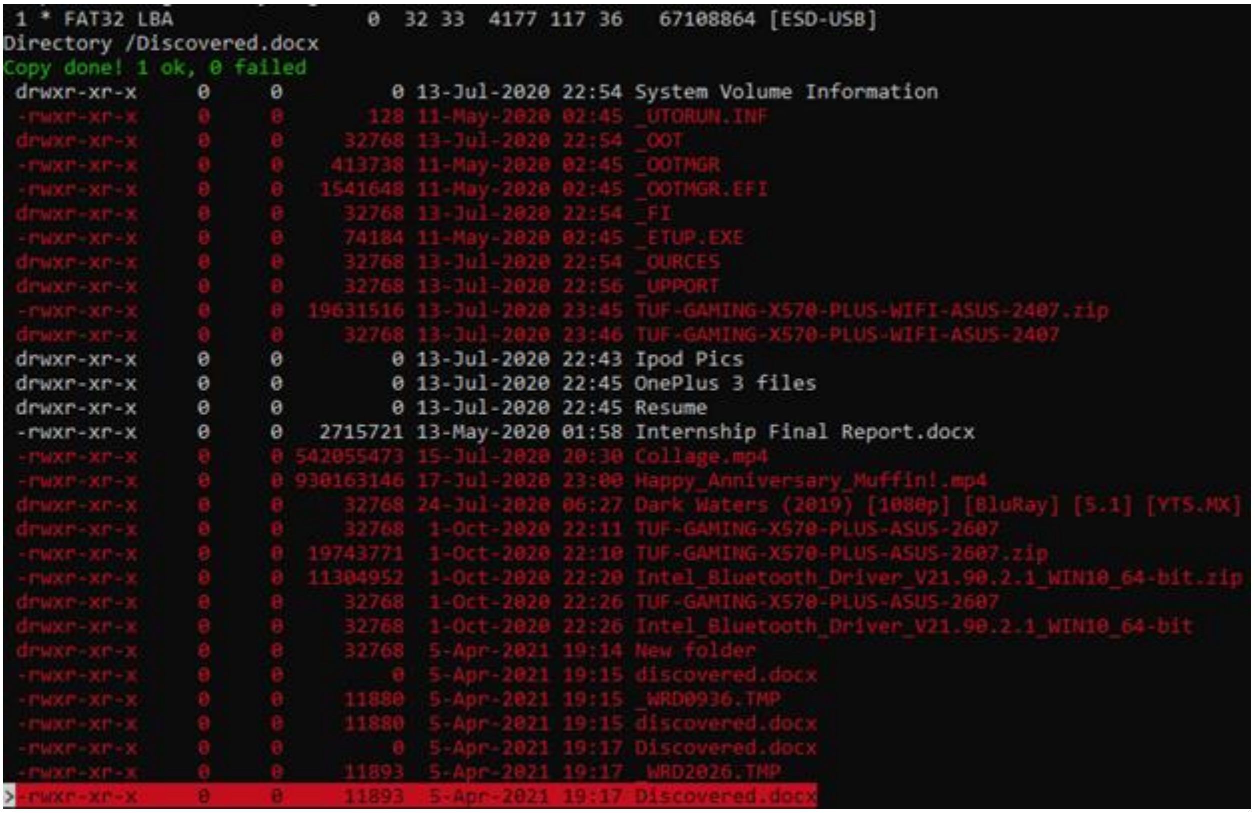
Once we are in the list of partition, we can see all the files and folders that have been deleted in the USB. The red data indicates the deleted files or folders. At the very bottom, we can see the “Discovered.docx” file that we have deleted. At the bottom of this compile, we can see the options we can utilize to do specific things. Since we only want to recover the Discovered.docx file, we well hover over it and press ‘c’ on the keyboard to store it in the clipboard.



```
TestDisk 7.2-WIP, Data Recovery Utility, March 2021

Please select a destination where /Discovered.docx will be copied.
Keys: Arrow keys to select another directory
      C when the destination is correct
      Q to quit
Directory C:\Users\Imad\Downloads
>drwx------ 197609 197121         0  4-Apr-2021 18:13 .
 drwx------ 197609 197121         0  4-Apr-2021 18:13 ..
 dr-xr-xr-x 197609 197121         0  5-Apr-2021 20:24 cygdrive
 dr-xr-xr-x 197609 197121         0  5-Apr-2021 20:24 dev
 dr-xr-xr-x 197609 197121         0  5-Apr-2021 20:24 proc
 drwx------ 197609 197121         0  5-Apr-2021 19:08 testdisk-7.2-WIP
```

Figure 1.14: The destination you wish to store the file.

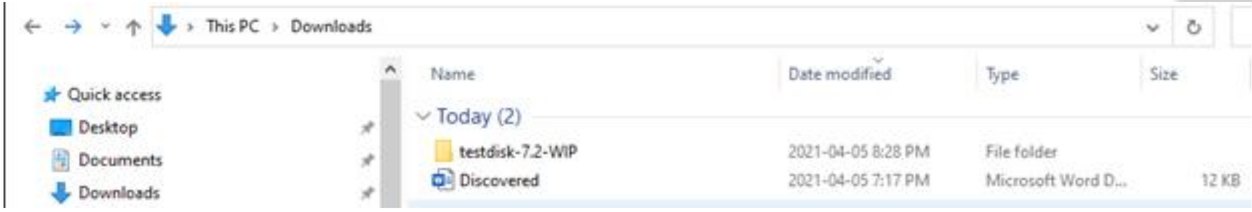
Once we copy it onto the clipboard, we must select a destination in our local computer to store the file. I will select my Download folder in my regular OS hard drive for example. Once we select the folder we wish to extract it in, we can type ‘shift+c’ in order to store it in the destination.



```
 1 * FAT32 LBA                0  32 33  4177 117 36   67108864 [ESD-USB]
Directory /Discovered.docx
Copy done! 1 ok, 0 failed
 drwxr-xr-x     0     0          0 13-Jul-2020 22:54 System Volume Information
 -rwxr-xr-x     0     0        128 11-May-2020 02:45 _UTORUN.INF
 drwxr-xr-x     0     0      32768 13-Jul-2020 22:54 _OOT
 -rwxr-xr-x     0     0     413738 11-May-2020 02:45 _OOTMGR
 -rwxr-xr-x     0     0    1541648 11-May-2020 02:45 _OOTMGR.EFI
 drwxr-xr-x     0     0      32768 13-Jul-2020 22:54 _FI
 -rwxr-xr-x     0     0      74184 11-May-2020 02:45 _ETUP.EXE
 drwxr-xr-x     0     0      32768 13-Jul-2020 22:54 _OURCES
 drwxr-xr-x     0     0      32768 13-Jul-2020 22:56 _UPPORT
 -rwxr-xr-x     0     0   19631516 13-Jul-2020 23:45 TUF-GAMING-X570-PLUS-WIFI-ASUS-2407.zip
 drwxr-xr-x     0     0      32768 13-Jul-2020 23:46 TUF-GAMING-X570-PLUS-WIFI-ASUS-2407
 drwxr-xr-x     0     0          0 13-Jul-2020 22:43 Ipod Pics
 drwxr-xr-x     0     0          0 13-Jul-2020 22:45 OnePlus 3 files
 drwxr-xr-x     0     0          0 13-Jul-2020 22:45 Resume
 -rwxr-xr-x     0     0    2715721 13-May-2020 01:58 Internship Final Report.docx
 -rwxr-xr-x     0     0  542055473 15-Jul-2020 20:30 Collage.mp4
 -rwxr-xr-x     0     0  930163146 17-Jul-2020 23:00 Happy_Anniversary_Muffin!.mp4
 drwxr-xr-x     0     0      32768 24-Jul-2020 06:27 Dark Waters (2019) [1080p] [BluRay] [5.1] [YTS.MX]
 drwxr-xr-x     0     0      32768  1-Oct-2020 22:11 TUF-GAMING-X570-PLUS-ASUS-2607
 -rwxr-xr-x     0     0   19743771  1-Oct-2020 22:10 TUF-GAMING-X570-PLUS-ASUS-2607.zip
 -rwxr-xr-x     0     0   11304952  1-Oct-2020 22:20 Intel_Bluetooth_Driver_V21.90.2.1_WIN10_64-bit.zip
 drwxr-xr-x     0     0      32768  1-Oct-2020 22:26 TUF-GAMING-X570-PLUS-ASUS-2607
 drwxr-xr-x     0     0      32768  1-Oct-2020 22:26 Intel_Bluetooth_Driver_V21.90.2.1_WIN10_64-bit
 drwxr-xr-x     0     0      32768  5-Apr-2021 19:14 New folder
 -rwxr-xr-x     0     0          0  5-Apr-2021 19:15 discovered.docx
 -rwxr-xr-x     0     0      11880  5-Apr-2021 19:15 _WRD0936.TMP
 -rwxr-xr-x     0     0      11880  5-Apr-2021 19:15 discovered.docx
 -rwxr-xr-x     0     0          0  5-Apr-2021 19:17 Discovered.docx
 -rwxr-xr-x     0     0      11893  5-Apr-2021 19:17 _WRD2026.TMP
>-rwxr-xr-x     0     0      11893  5-Apr-2021 19:17 Discovered.docx
```

Figure 1.15: Confirmation that the file has been stores successfully.

Once TestDisk brings you back to the list of files and folders again with the following message with 0 failed, then the file has been stored in the destination successfully.



Name	Date modified	Type	Size
Today (2)			
testdisk-7.2-WIP	2021-04-05 8:28 PM	File folder	
Discovered	2021-04-05 7:17 PM	Microsoft Word D...	12 KB

Figure 1.16: verifying the file has been stored in the correct destination

We can see the file has been stored successfully in the Download folder in my C: solid state drive. From here, the file is a regular file and you could open it, change it and edit it as you please.

3 - The Sleuth Kit (TSK)

3.1 - What is The Sleuth Kit

The Sleuth Kit (TSK) is an open-source software that has a library and collection of command lines that you could do on many different operating systems to investigate and analyze volume and file system data [3]. TSK can identify and recover evidence from files on hard drives which can be used directly to find criminal evidence [3]. In addition, TSK allows users to analyze a disk or file system created by ‘dd’ files, which is good for practice for an IT forensic user. TSK has features that are classified as low-level data extraction to high-level data extraction.

3.2 - My Opinion about The Sleuth Kit

TSK is great because it has many features and multiple different options built in for analyzing and recovering data. The sleuth kit’s official website is not user friendly, and it takes a little bit of browsing in order to get familiar with the interface. However, once you get accustomed with the website, you can find an abundance of information regarding the functionality and commands that are related to TSK.

3.3 - How does it work?

TSK uses a mixture of many different libraries and commands to do various things with file systems. TSK can recover whole directories and can recover many different file formats which include:

- JPEG
- pdf
- PNG
- doc
- txt

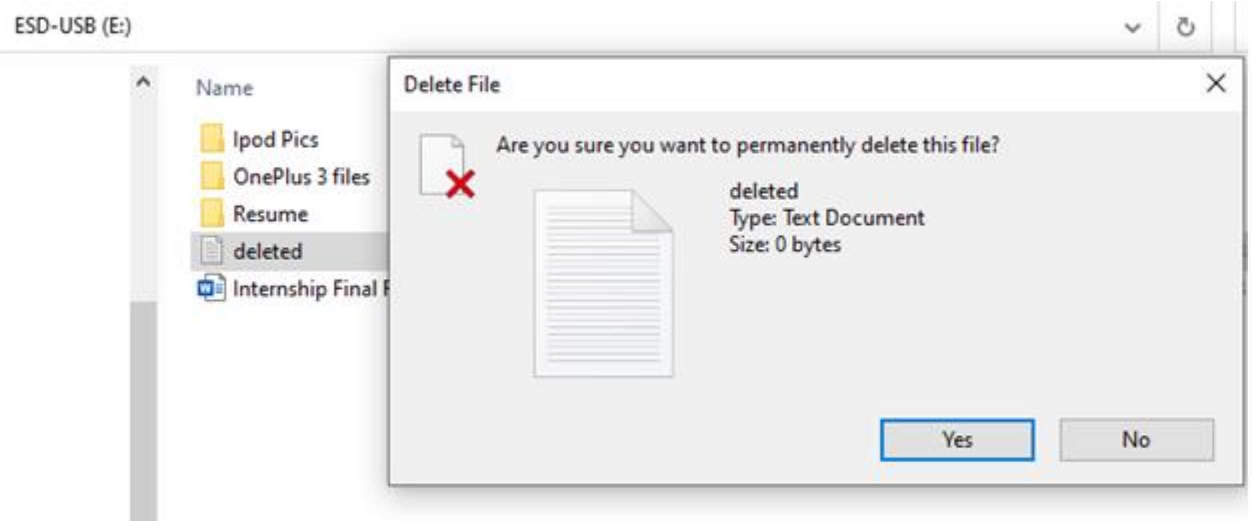
We will be going in-depth and a live demo showcasing how TSK works (see live demo: Sleuth Kit). This section will give a general understanding of how TSK functions. Before we start the process of analyzing and recovering files, it is important to first gain information about the USB itself.

1. The first step is to do a “mmls” command to see the different partitions in the device.
2. Once we find the right partition we want to enter, we need to extract the first offset numbers of the specified partition so we can utilize it in our “fls” command which will be used to list all files in the device.
3. Next, the fls command lists the deleted and stored folders/files in the device.
4. “icat” is a great command to utilize since it shows the content of the file without the need of recovering the file onto a local drive.
5. Lastly, “tsk_recover” is a command that recovers folders/files onto a local drive.

This is a very high-end explanation of how TSK extracts data. To understand more about TSK and how it functions please go to section xxx. (live demo: sleuth kit)

3.4 - Live Demo: The Sleuth Kit

For this live demo we will be analyzing the content in the .txt file that we have deleted. In addition, we will be extracting it so that it is no longer deleted. This will explain how files in TSK can recover deleted files.

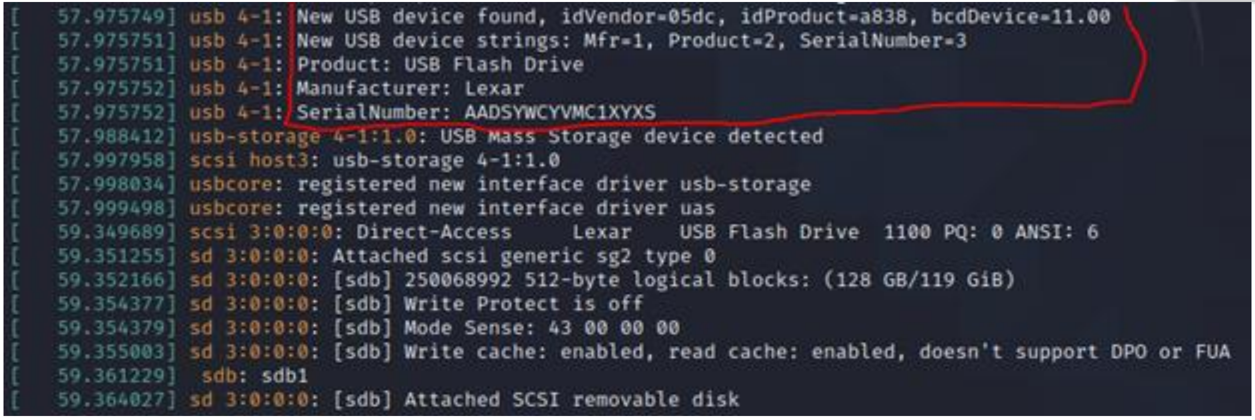


Are you sure you want to permanently delete this file?

deleted
Type: Text Document
Size: 0 bytes

Figure 2.1: The sample deleted.txt test we will be using

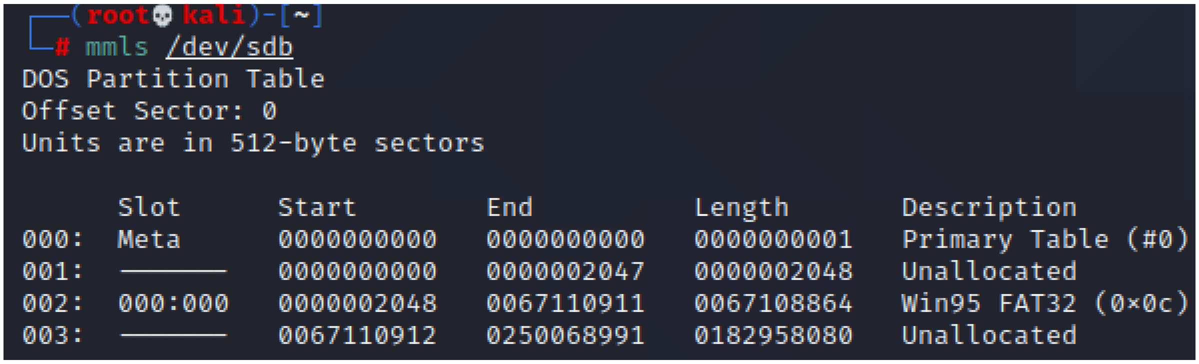
We need to first delete the sample .txt file we are using in this live demo. The file that we are using is named “deleted.txt”.



```
[   57.975749] usb 4-1: New USB device found, idVendor=05dc, idProduct=a838, bcdDevice=11.00
[   57.975751] usb 4-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   57.975751] usb 4-1: Product: USB Flash Drive
[   57.975752] usb 4-1: Manufacturer: Lexar
[   57.975752] usb 4-1: SerialNumber: AADSYWCYVMC1XYXS
[   57.988412] usb-storage 4-1:1.0: USB Mass Storage device detected
[   57.997958] scsi host3: usb-storage 4-1:1.0
[   57.998034] usbcore: registered new interface driver usb-storage
[   57.999498] usbcore: registered new interface driver uas
[   59.349689] scsi 3:0:0:0: Direct-Access     Lexar    USB Flash Drive  1100 PQ: 0 ANSI: 6
[   59.351255] sd 3:0:0:0: Attached scsi generic sg2 type 0
[   59.352166] sd 3:0:0:0: [sdb] 250068992 512-byte logical blocks: (128 GB/119 GiB)
[   59.354377] sd 3:0:0:0: [sdb] Write Protect is off
[   59.354379] sd 3:0:0:0: [sdb] Mode Sense: 43 00 00 00
[   59.355003] sd 3:0:0:0: [sdb] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
[   59.361229]  sdb: sdb1
[   59.364027] sd 3:0:0:0: [sdb] Attached SCSI removable disk
```

Figure 2.2: The messages of the USB being found in kali

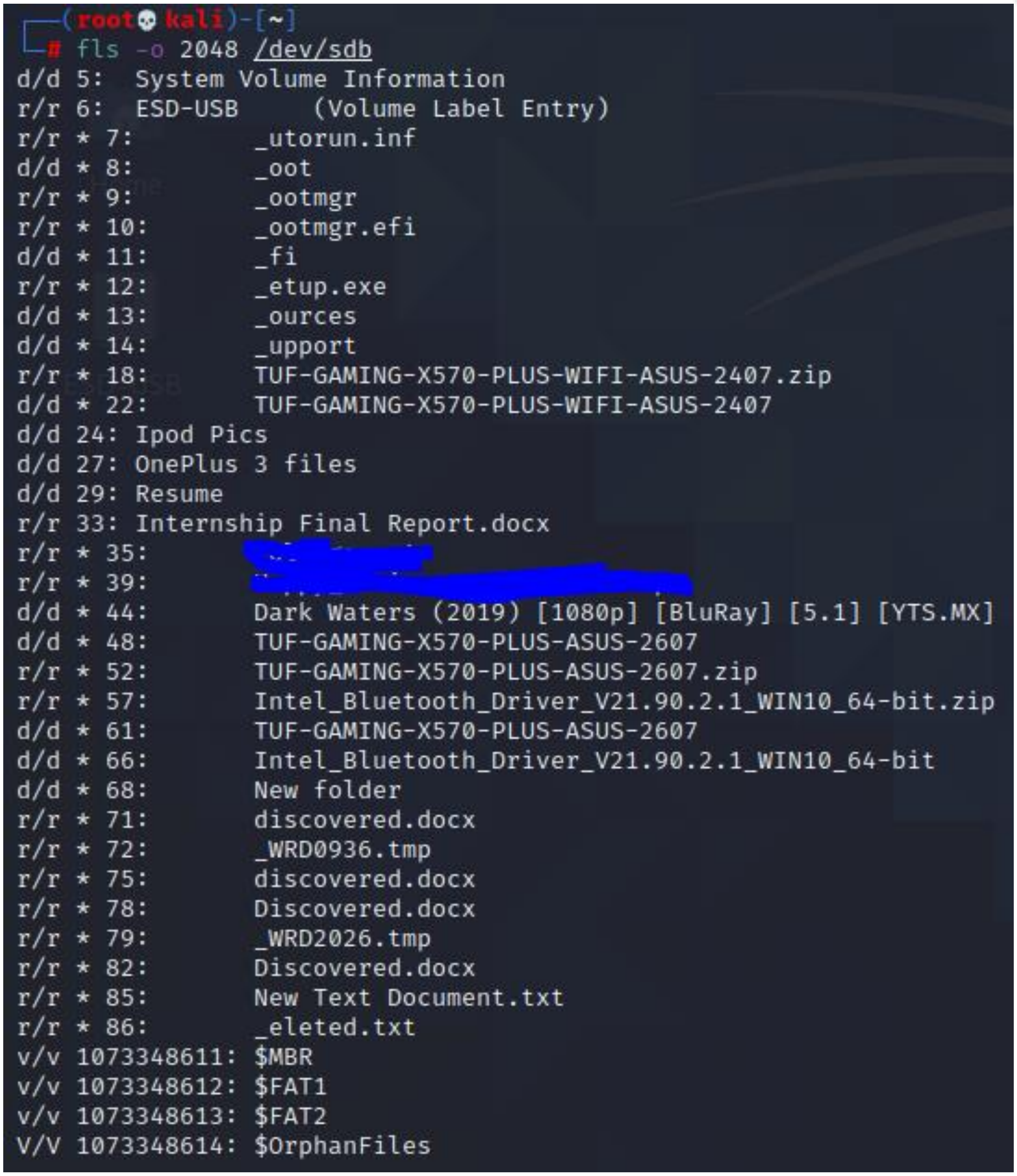
Since we are using kali on a virtual machine, we need to verify if the USB has been captured in our virtual machine in kali. By doing a command “dmesg” we can see that kali has installed the USB and given it a device number which is sdb1 (see *figure 2.2*).



```
┌──(root💀 kali)-[~]
└─# mmls /dev/sdb
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0067110911   0067108864   Win95 FAT32 (0×0c)
003:  -------   0067110912   0250068991   0182958080   Unallocated
```

Figure 2.3: the different sections of partition in the USB

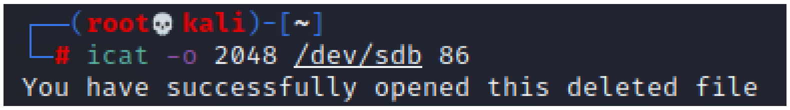
We need to write the command “mmls” followed by where the USB has been installed. By default, Linux installs and stores its devices in the /dev path. The reason we need to do this output is check which partition are files in. In *figure 2.3* we can see that this USB is a FAT file system, and we can see the offset start and end numbers. We need to copy the start offset number for our next command.



```
┌──(root💀kali)-[~]
└─# fls -o 2048 /dev/sdb
d/d 5:  System Volume Information
r/r 6:  ESD-USB     (Volume Label Entry)
r/r * 7:        _utorun.inf
d/d * 8:        _oot
r/r * 9:        _ootmgr
r/r * 10:       _ootmgr.efi
d/d * 11:       _fi
r/r * 12:       _etup.exe
d/d * 13:       _ources
d/d * 14:       _upport
r/r * 18:       TUF-GAMING-X570-PLUS-WIFI-ASUS-2407.zip
d/d * 22:       TUF-GAMING-X570-PLUS-WIFI-ASUS-2407
d/d 24: Ipod Pics
d/d 27: OnePlus 3 files
d/d 29: Resume
r/r 33: Internship Final Report.docx
r/r * 35:       [redacted]
r/r * 39:       [redacted]
d/d * 44:       Dark Waters (2019) [1080p] [BluRay] [5.1] [YTS.MX]
d/d * 48:       TUF-GAMING-X570-PLUS-ASUS-2607
r/r * 52:       TUF-GAMING-X570-PLUS-ASUS-2607.zip
r/r * 57:       Intel_Bluetooth_Driver_V21.90.2.1_WIN10_64-bit.zip
d/d * 61:       TUF-GAMING-X570-PLUS-ASUS-2607
d/d * 66:       Intel_Bluetooth_Driver_V21.90.2.1_WIN10_64-bit
d/d * 68:       New folder
r/r * 71:       discovered.docx
r/r * 72:       _WRD0936.tmp
r/r * 75:       discovered.docx
r/r * 78:       Discovered.docx
r/r * 79:       _WRD2026.tmp
r/r * 82:       Discovered.docx
r/r * 85:       New Text Document.txt
r/r * 86:       _eleted.txt
v/v 1073348611: $MBR
v/v 1073348612: $FAT1
v/v 1073348613: $FAT2
V/V 1073348614: $OrphanFiles
```

Figure 2.4: List of all files and folders that were deleted and stored

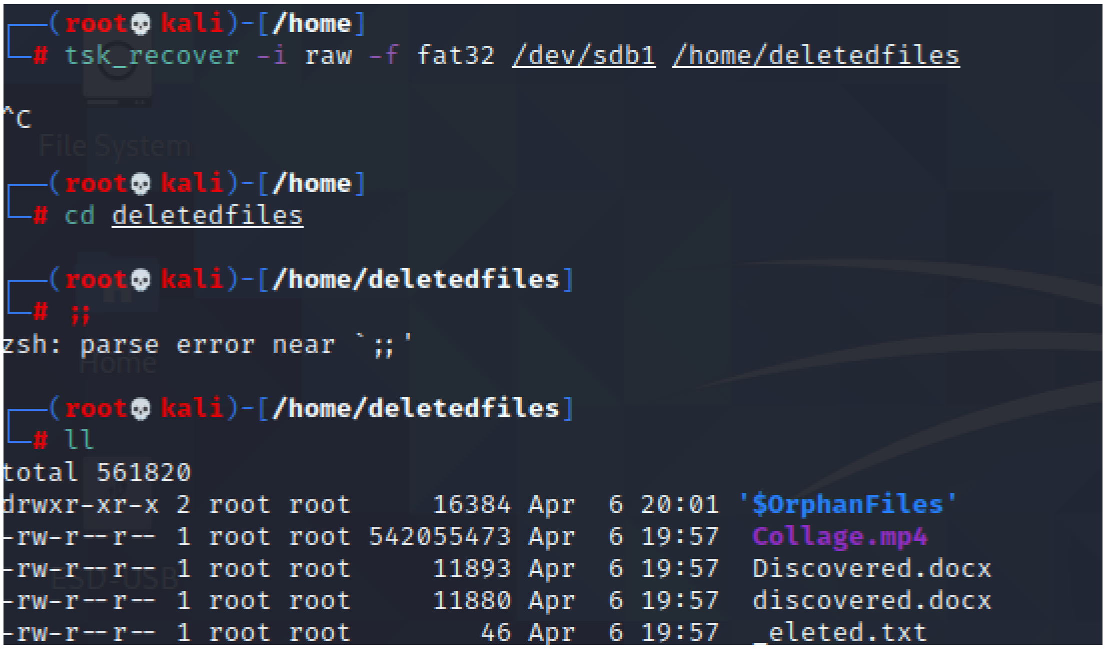
Once we do the command “fls -o 2048 /dev/sdb” we can see all the file lists and folders in the USB. Whether they are deleted or still in storage. Fls is the command for file list, -o is to specify the offset start number of the partition, and /dev/sdb is specifying which device is being analyzed. “r/r” represent regular files while “d/d” represent directories. Every file and directory that has an asterisk beside it means that it is deleted. All the other files and folders that do not mean that they are still being kept in storage. Once we verify that our deleted .txt file is in here which it is. We can start the analyzing and extraction process. We need to keep in mind the inumber for the next command. The inumber is the unique number each directory and file has beside it which is used to enter the directory or analyze the content of files. For example the inumber for deleted.txt file is 86 (*see figure 2.4*).



```
┌──(root💀kali)-[~]
└─# icat -o 2048 /dev/sdb 86
You have successfully opened this deleted file
```

Figure 2.5: shows the content of the deleted.txt file

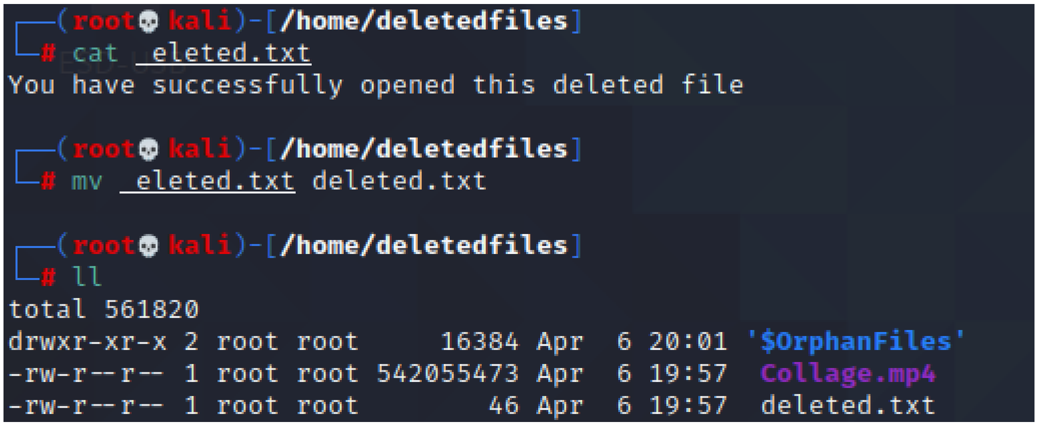
“icat” is a command that allows us to analyze the data in the file and in *figure 2.5* we can see the content of the file.



```
┌──(root💀kali)-[/home]
└─# tsk_recover -i raw -f fat32 /dev/sdb1 /home/deletedfiles

^C

┌──(root💀kali)-[/home]
└─# cd deletedfiles

┌──(root💀kali)-[/home/deletedfiles]
└─# ;;
zsh: parse error near `;;'

┌──(root💀kali)-[/home/deletedfiles]
└─# ll
total 561820
drwxr-xr-x 2 root root     16384 Apr  6 20:01 '$OrphanFiles'
-rw-r--r-- 1 root root 542055473 Apr  6 19:57  Collage.mp4
-rw-r--r-- 1 root root     11893 Apr  6 19:57  Discovered.docx
-rw-r--r-- 1 root root     11880 Apr  6 19:57  discovered.docx
-rw-r--r-- 1 root root        46 Apr  6 19:57  _eleted.txt
```

Figure 2.6: recovering the deleted files

We can take this one step further by extracting all the files and folders into kali. In *figure 2.6* I showcase how we can use the command tsk_recover to recover all deleted files into my /home/deletedfiles path using the syntax -i for the specific image type and -f for the specific file system type.



```
┌──(root💀kali)-[/home/deletedfiles]
└─# cat _eleted.txt
You have successfully opened this deleted file

┌──(root💀kali)-[/home/deletedfiles]
└─# mv _eleted.txt deleted.txt

┌──(root💀kali)-[/home/deletedfiles]
└─# ll
total 561820
drwxr-xr-x 2 root root     16384 Apr  6 20:01 '$OrphanFiles'
-rw-r--r-- 1 root root 542055473 Apr  6 19:57  Collage.mp4
-rw-r--r-- 1 root root        46 Apr  6 19:57  deleted.txt
```

Figure 2.7: showcasing the recovered files

Once we finish recovering the files, they become normal files in kali. To showcase this, I showed the content of the deleted.txt file again and also changed the name to verify it is a regular file.

4 - TestDisk vs. The Sleuth Kit (TSK)

Both softwares are open sourced which is good for users since they do not need to pay for it. TestDisk has a pdf file on its official website which contains step-by step instructions of its features and all its functionalities. Sleuth kit has very detailed documentation too however it is scattered in multiple documentations. This is good as it has more features then TestDisk however, it is harder to learn all the content and figure out how to do a specific feature as you will be likely flipping through multiple documentation to get all the information you are looking for.

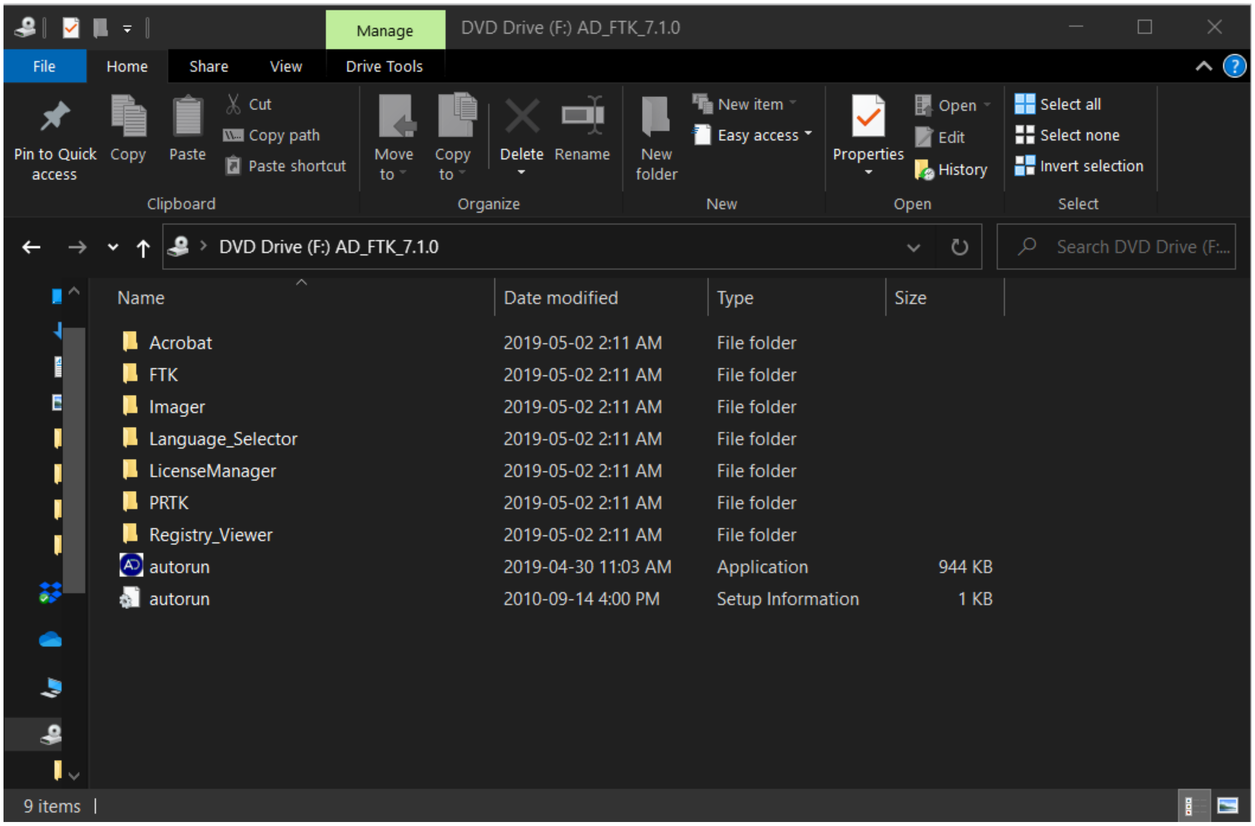
TSK is much more advanced and is suited for people who want to become an advanced user in the field of IT forensics. Things such as recovering all files and folders from a device is much easier to do on TSK because of its wide variety of features. Another example would be the ease of accessing files and folders without the need of extracting and recovering the data. A drawback of TSK is that it is more complicated to use on Windows since it has additional installation to do rather then just the TSK software itself.

TestDisk works on multiple different operating systems and is easily installed from the official website with one click. Different operating systems do not require additional steps unlike TSK. TestDisk is more user-friendly and is easier to grasp.

In conclusion, TestDisk and TSK are both very good and I would recommend the two different softwares depending on who the user is and how skilled they are in regard to IT. They both have their advantages and disadvantages. While TestDisk is better for beginner users learning data recovery, sleuth kit is better for advanced users who are well adapted with IT forensics. In addition, TSK is better for users who are attempting to get better at IT forensics and want to become more advanced as an IT forensic user.

5 - AccessData Forensic Toolkit

After downloading this software, you are left with an .iso file that contains many different and unique programs under the toolkit. It can be confusing for first time users to find and install the desired program that is needed.



Name	Date modified	Type	Size
Acrobat	2019-05-02 2:11 AM	File folder	
FTK	2019-05-02 2:11 AM	File folder	
Imager	2019-05-02 2:11 AM	File folder	
Language_Selector	2019-05-02 2:11 AM	File folder	
LicenseManager	2019-05-02 2:11 AM	File folder	
PRTK	2019-05-02 2:11 AM	File folder	
Registry_Viewer	2019-05-02 2:11 AM	File folder	
autorun	2019-04-30 11:03 AM	Application	944 KB
autorun	2010-09-14 4:00 PM	Setup Information	1 KB

Fig 3.1. Shows the screen after opening the .iso file.

After finding the correct program we needed to install we were greeted with the following user interface. It has a fairly simple design that is not overcrowded and it’s easy to follow. Most users will have a pleasant experience and can easily find different tools with this interface.

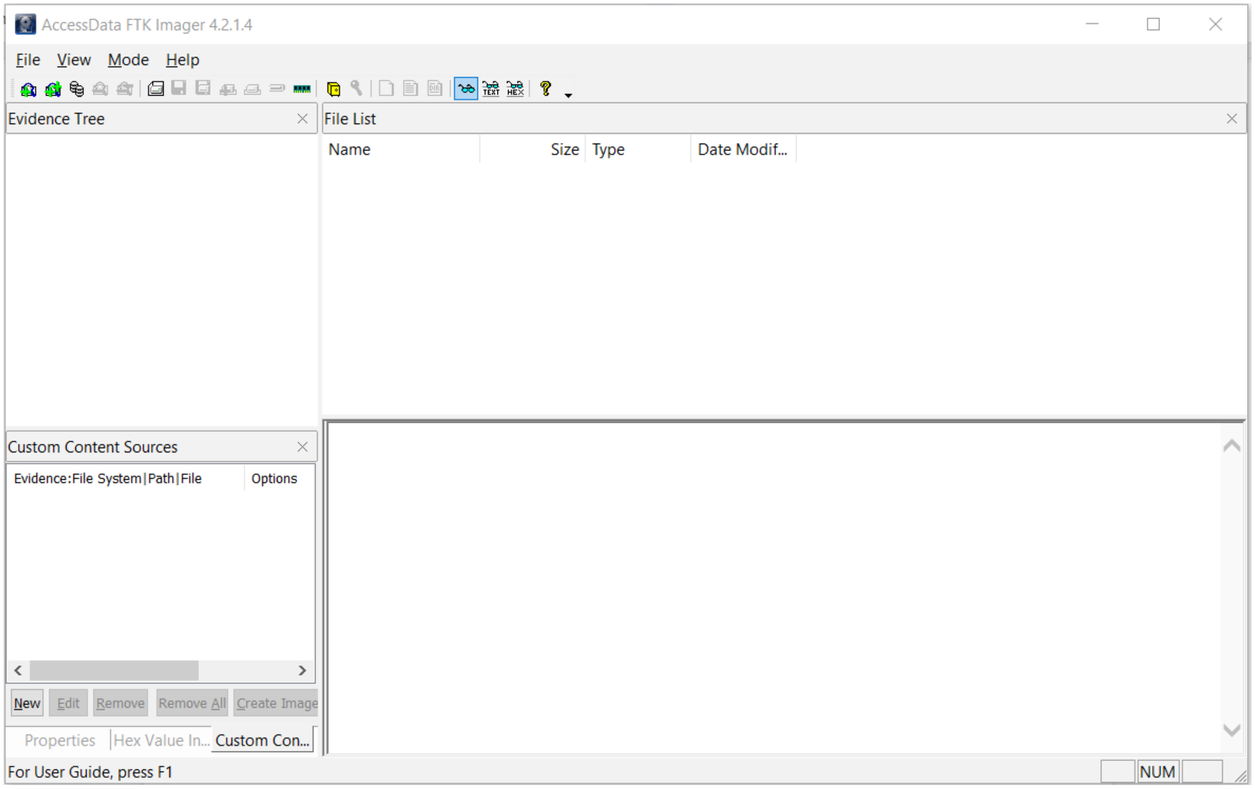
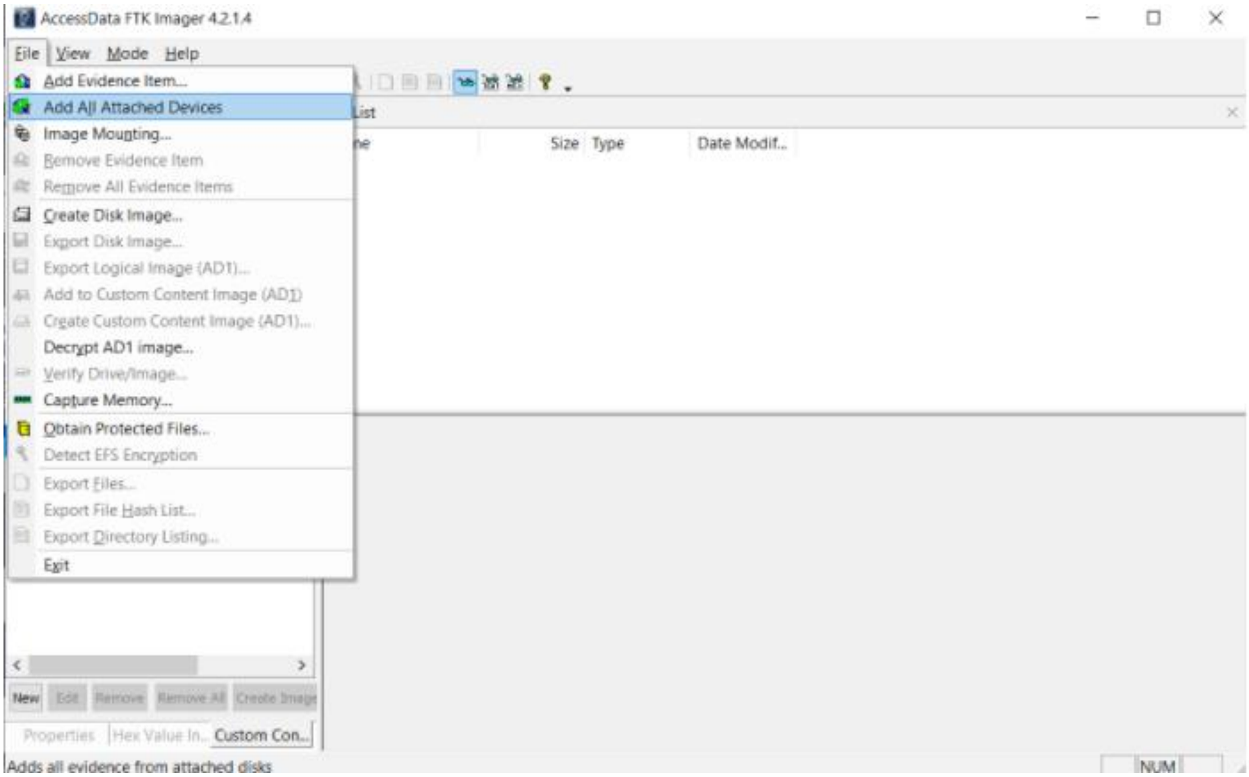


Fig 3.2. Shows the user interface of the Forensic Toolkit Imager program.

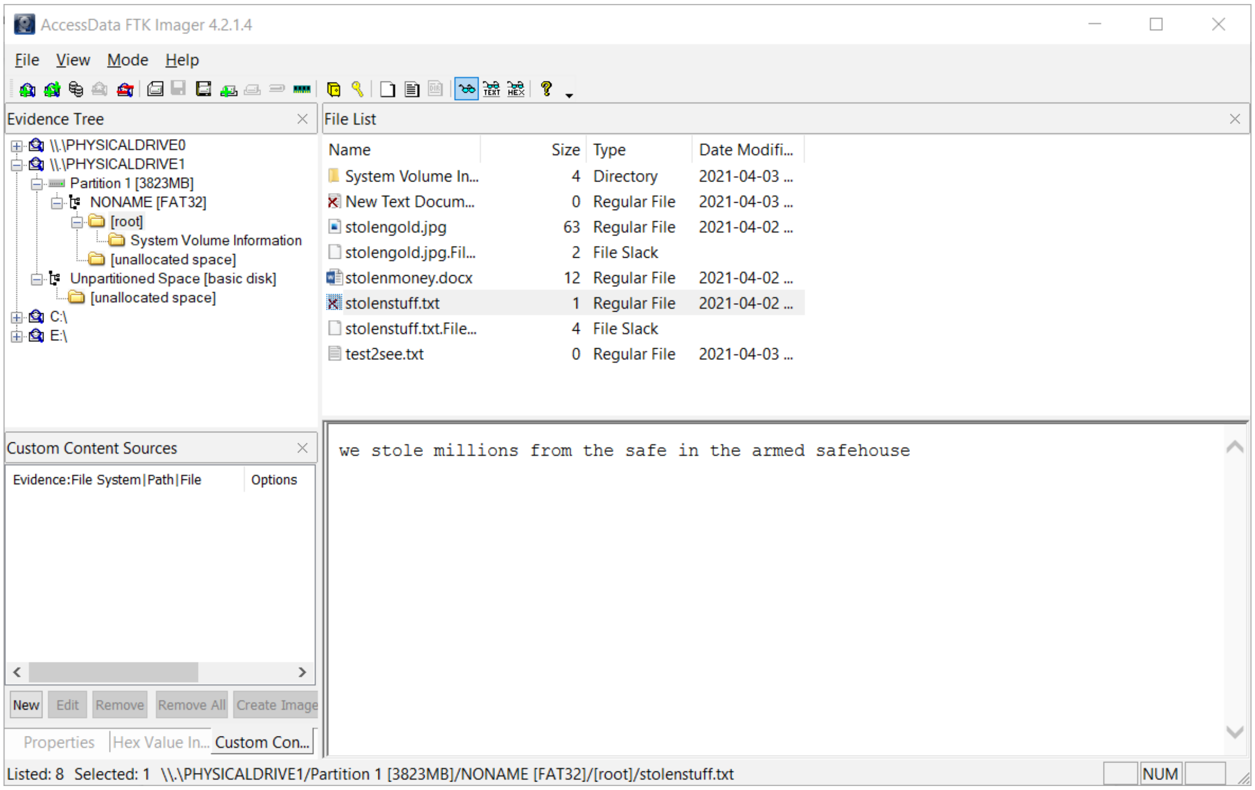
Now a demo will be performed on how to recover deleted files from a USB drive. Once the Forensics Toolkit program opens a user can simply click on “Add Evidence Item” to attach a single device or media source or they can select “Add All Attached Devices” which adds all drives very quickly to the system and the user can simply locate the desired drive, this is the method we opted to use for this example.



Adds all evidence from attached disks

Fig 3.3. Shows how to mount drives or media sources.

After adding all the devices/media source to the Forensic Toolkit program we navigated through the devices and located the 4GB USB flash drive. Expanding all the nodes on the left-hand side of the interface will quickly allow you to find where the files that are stored on the USB flash drive are on the toolkit. *Figure 3.4* shows the files that are on this 4GB USB flash drive, the files with a small red "x" on the icons are files that have been deleted. By simply clicking on one of the deleted files you can reveal its contents as shown in the figure below. The delete files can also be exported by right-clicking them and selection "Export Files" to any desired location.



Name	Size	Type	Date Modifi...
System Volume In...	4	Directory	2021-04-03 ...
New Text Docum...	0	Regular File	2021-04-03 ...
stolengold.jpg	63	Regular File	2021-04-02 ...
stolengold.jpg.Fil...	2	File Slack	
stolenmoney.docx	12	Regular File	2021-04-02 ...
stolenstuff.txt	1	Regular File	2021-04-02 ...
stolenstuff.txt.File...	4	File Slack	
test2see.txt	0	Regular File	2021-04-03 ...

```
we stole millions from the safe in the armed safehouse
```

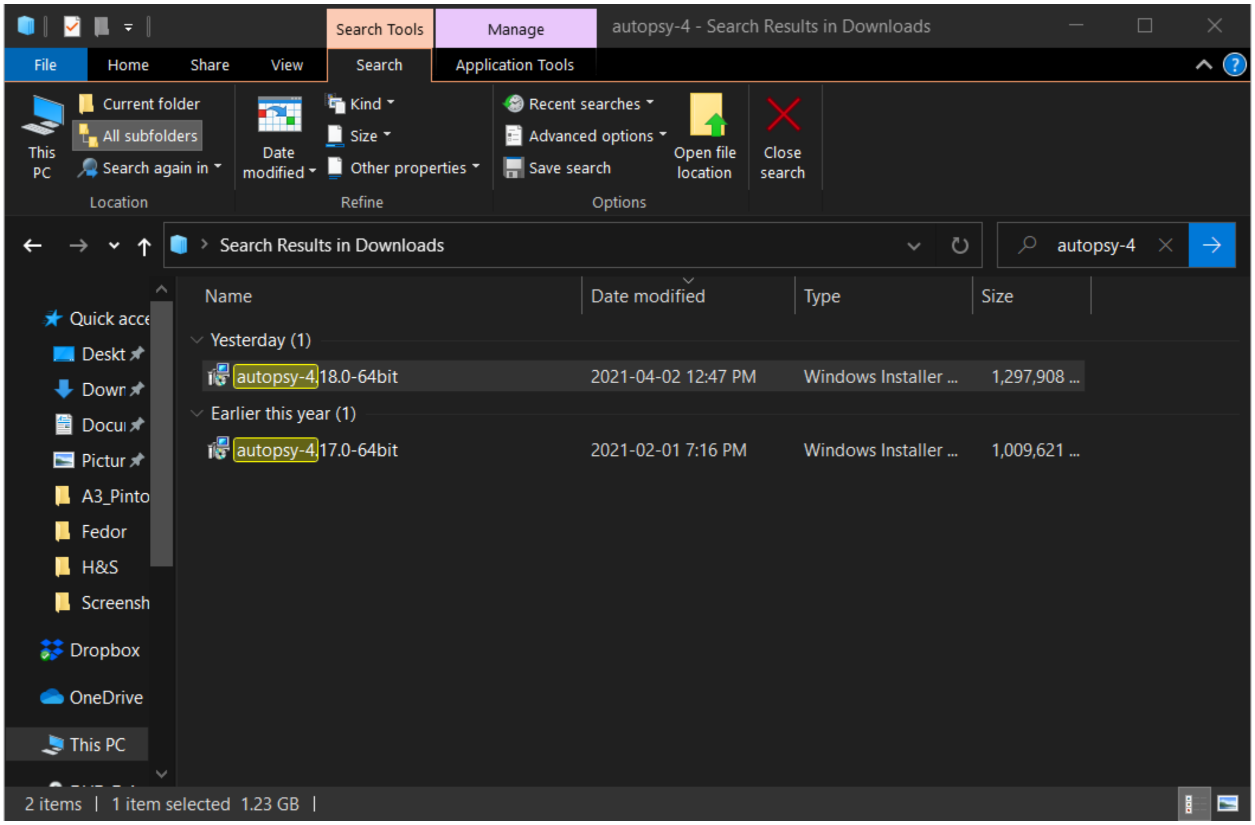
Listed: 8 Selected: 1 \\.\PHYSICALDRIVE1/Partition 1 [3823MB]/NONAME [FAT32]/[root]/stolenstuff.txt

Fig 3.4. Shows the contents and deleted files on the USB flash drive.

Overall, we would recommend the Forensic Toolkit to anyone looking to recover deleted files. The positives of this program are that you can quickly attach/add devices/media sources, open them up, and find what you're looking for. A huge plus is that it will show the deleted files with the other files by including a red “x” on the icon. This saves the end user plenty of time as no other steps are required to locate deleted files and then attempt to reveal that files contents. The only downside to this toolkit is that when you first download the .iso file it can be overwhelming and complicated on deciding which program you need to install as there are many to choose from.

6 - Autopsy

After downloading Autopsy you are given a Windows Installer file that guides you through the entire installation. Once installed you can open up the Autopsy.exe file which is the Autopsy program/software itself. It is a very easy process that would be difficult to mess up.



Name	Date modified	Type	Size
Yesterday (1)			
autopsy-4.18.0-64bit	2021-04-02 12:47 PM	Windows Installer ...	1,297,908 ...
Earlier this year (1)			
autopsy-4.17.0-64bit	2021-02-01 7:16 PM	Windows Installer ...	1,009,621 ...

Fig 4.1. Shows the file downloaded to install Autopsy.

Once you open Autopsy for the first time you are greeted by a vicious looking Doberman holding a magnifying glass. It's a very simple design that is easy to follow for a pleasant user experience. It also prompts users with the option to open a "New Case", "Open Recent Case", or "Open Case". Now a demo involving recovering deleted files from a USB flash drive will be shown. In this demo we are choosing to open a new case.

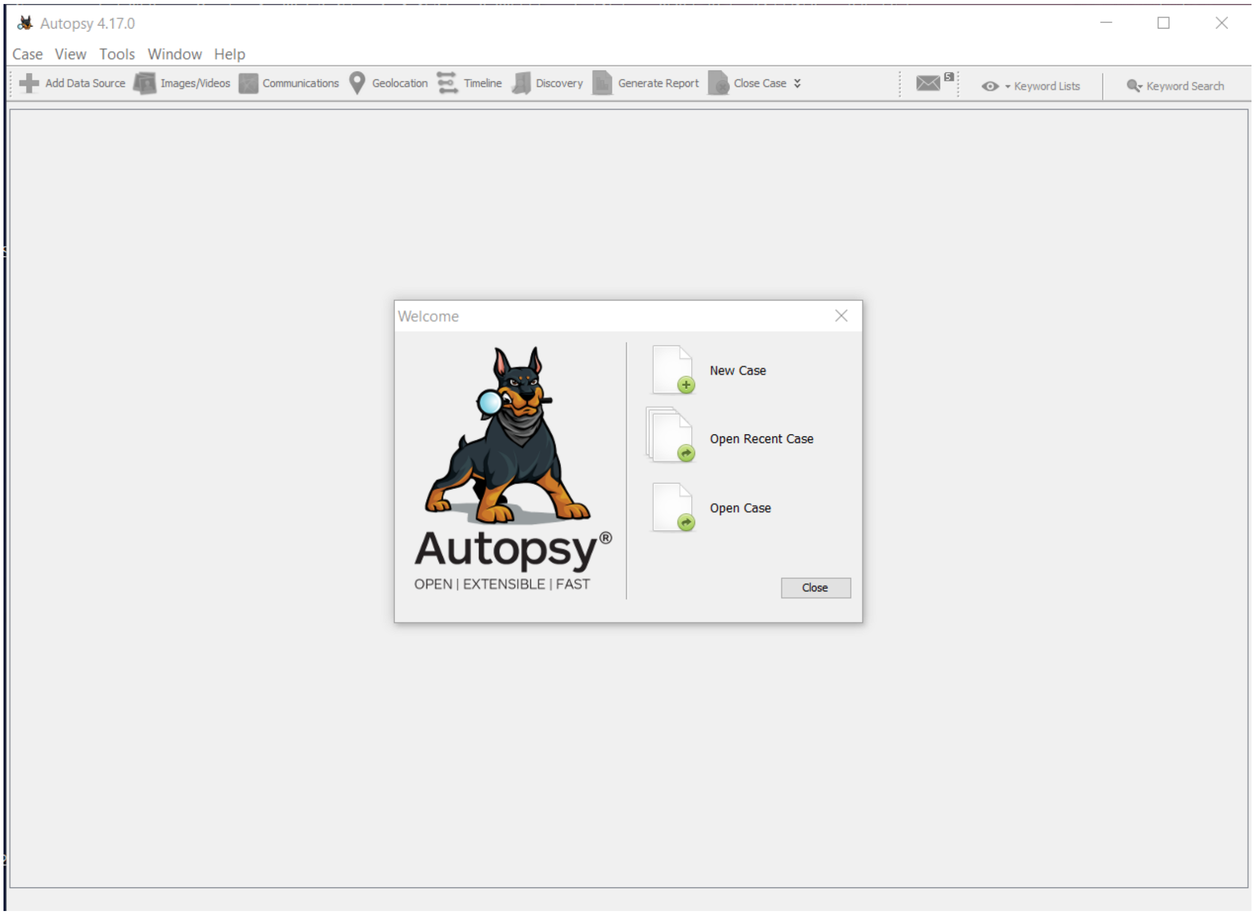
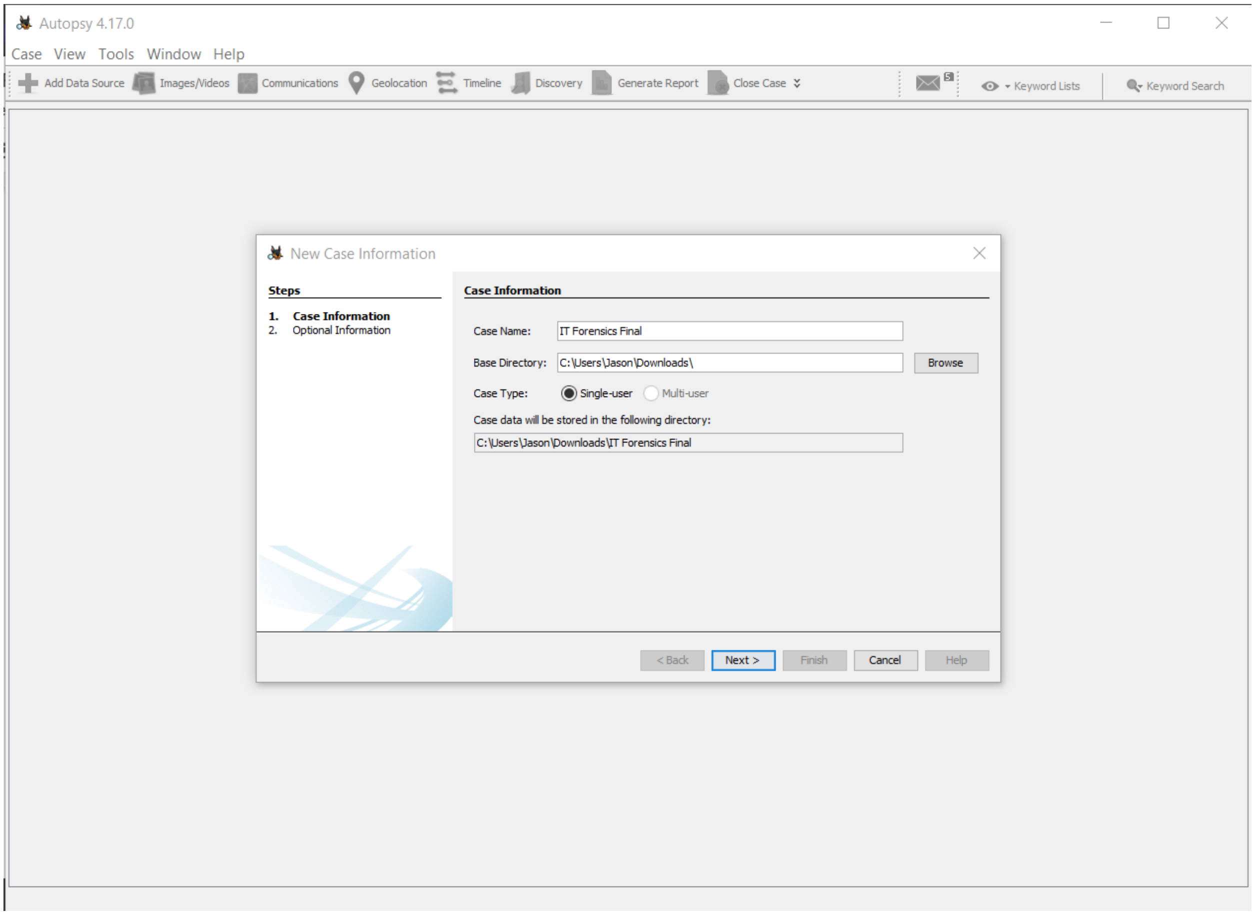


Fig 4.2. Shows the user interface on the Autopsy program.

After choosing to create a new case we are required to input some case information such as a case name. Here we simply input “IT Forensics Final” and then clicked “next” and then clicked “finish”, and our case was created.



Case Information

Case Name: IT Forensics Final

Base Directory: C:\Users\Jason\Downloads\

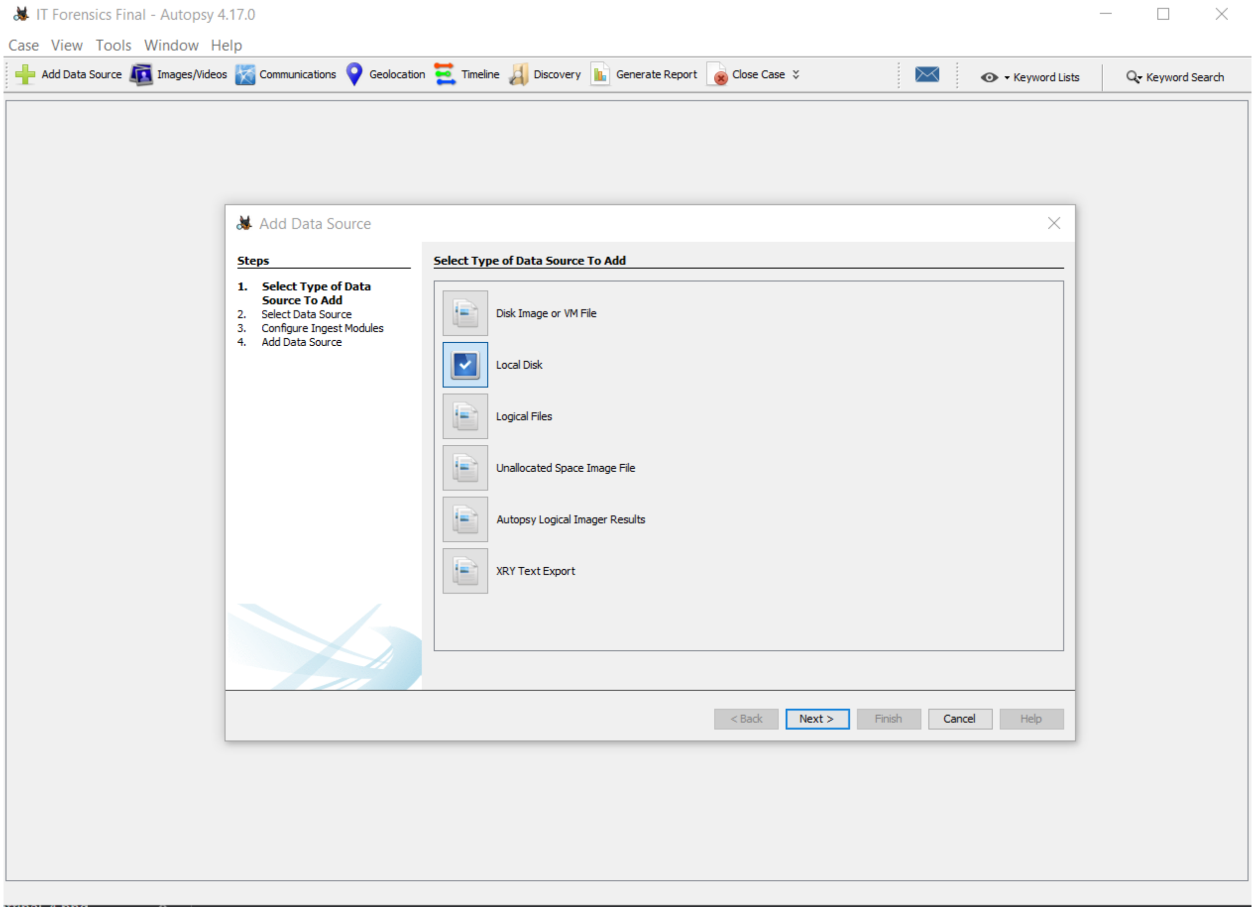
Case Type: ● Single-user ○ Multi-user

Case data will be stored in the following directory:

C:\Users\Jason\Downloads\IT Forensics Final

Fig 4.3. Shows how to create a “New Case”.

Once our case was created, we needed to input some data source, *figure 4.4* shows the many options a user will have when selecting which type of data they would like to import to Autopsy. In this scenario we are using a USB flash drive which is considered a local disk when inserted into a computer. After selecting the disk type, a user is prompted to select the disk itself, so we selected the 4GB USB flash drive, clicked “next” twice, and after our data source was processed and had been added to the local database we could then click “finish”.



Add Data Source

Steps

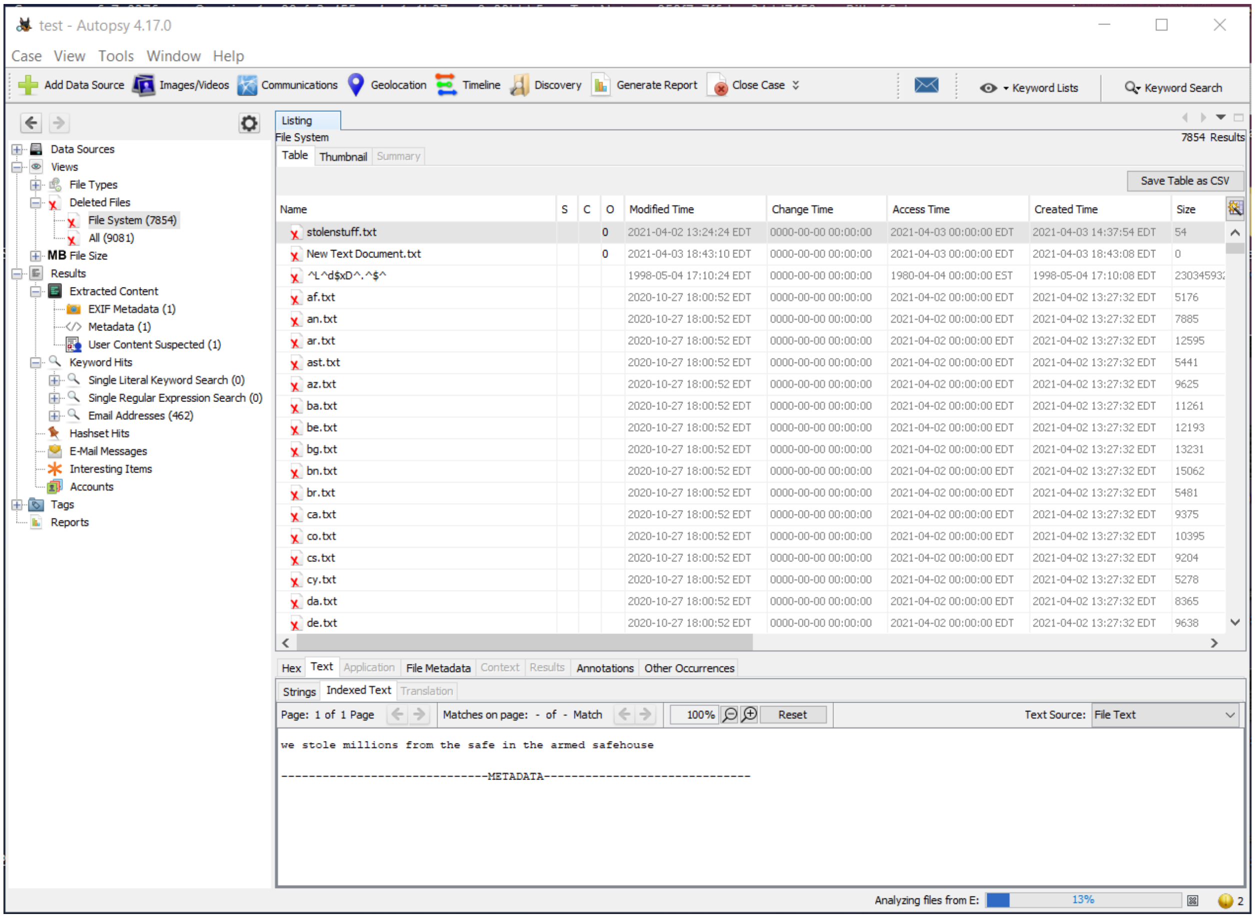
1. **Select Type of Data Source To Add**
2. Select Data Source
3. Configure Ingest Modules
4. Add Data Source

Select Type of Data Source To Add

- Disk Image or VM File
- ☑ Local Disk
- Logical Files
- Unallocated Space Image File
- Autopsy Logical Imager Results
- XRY Text Export

Fig 4.4. Shows the Data Source types a user can select.

After completing the above steps our 4GB USB flash drive was visible on the Autopsy database. Then we expanded the delete files node and clicked on “File Systems” there we could see all of the deleted files that were on this USB flash drive. The top 2 files as shown in *figure 4.5* are the files that were deleted from the flash drive by someone. The other files below are software files from formatting. By clicking on one of the deleted files “stolenstuff.txt” we were able to see exactly what the contents of that file were as shown.



Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size
stolenstuff.txt			0	2021-04-02 13:24:24 EDT	0000-00-00 00:00:00	2021-04-03 00:00:00 EDT	2021-04-03 14:37:54 EDT	54
New Text Document.txt			0	2021-04-03 18:43:10 EDT	0000-00-00 00:00:00	2021-04-03 00:00:00 EDT	2021-04-03 18:43:08 EDT	0
^L^d$xD^.^$^				1998-05-04 17:10:24 EDT	0000-00-00 00:00:00	1980-04-04 00:00:00 EST	1998-05-04 17:10:08 EDT	23034593
af.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	5176
an.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	7885
ar.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	12595
ast.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	5441
az.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	9625
ba.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	11261
be.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	12193
bg.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	13231
bn.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	15062
br.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	5481
ca.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	9375
co.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	10395
cs.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	9204
cy.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	5278
da.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	8365
de.txt				2020-10-27 18:00:52 EDT	0000-00-00 00:00:00	2021-04-02 00:00:00 EDT	2021-04-02 13:27:32 EDT	9638

```
we stole millions from the safe in the armed safehouse

-----------------------------METADATA-----------------------------
```

Fig 4.5. Shows the deleted files on the USB flash drive.

Overall, we would definitely recommend Autopsy for any user looking to recover deleted files from a USB flash drive. Autopsy does a great job of not only showing which files were deleted by someone, but also files that were deleted by the system when formatting, etc., but also places the files that were manually deleted at the top as these are generally the files a user would be searching for. The Autopsy interface is quite user friendly, it's very easy to follow along and it has great features such as being able to create new cases if a user would like to save their progress at any given point and resume at a later date. A huge plus is that it will show the deleted files in a separate section by including a red "x" on the icon. This saves the end user plenty of time as no other steps are required to locate deleted files and then attempt to reveal that files contents. The only downside compared to other forensic programs is that Autopsy requires users to create a new case or open an existing case and it can be time consuming before a data source is imported into the local database.

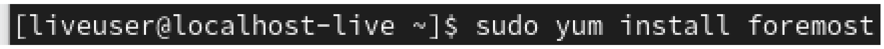
7 - Foremost

Foremost is a data recovery program that is based on Linux that uses file craving to recover data off a disk image. It was created in 2001 by the United States Air Force Office of Special

Investigations and The Center for Information Systems Security Studies and Research [4]. It initially was made for the US government, but later on, was opened to the public to have developers try to improve it. Foremost is one of the most well known data recovery programs that are available as many companies around the world use it to recover lost data from their disk drives. It uses the headers, footers, and other data structures of the files to recover them [4]. This means that it calculates the sectors that the deleted files are located in and then carves it out of the disk drive to recover it for the user. Foremost works on numerous file types and you can specify the file type in the command line when calling the program on a disk drive.

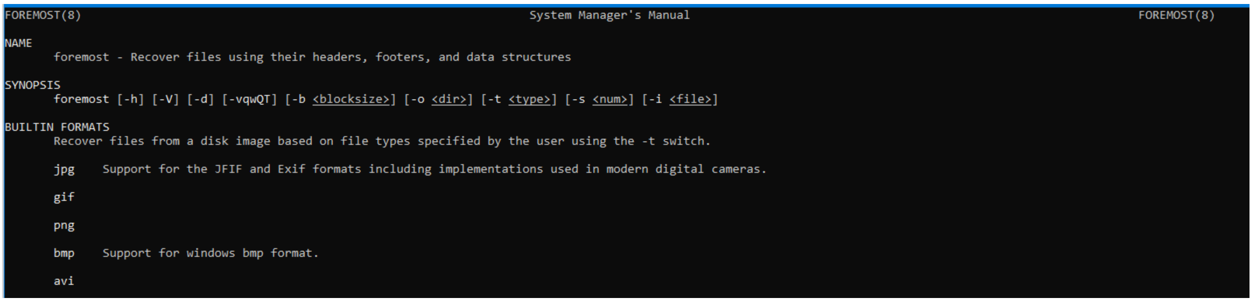
JPGs, GIFs, PNGs, EXE, MOV, PDF, MPG, and others are all supported by Foremost and it can understand the various headers of each file type to be able to carve it out. In this section, we will go into detail on the various functions of Foremost and show an example of it working.

To install Foremost, we can run the sudo yum install Foremost command on Fedora based Linux systems and that will install the required packages to run [4]:

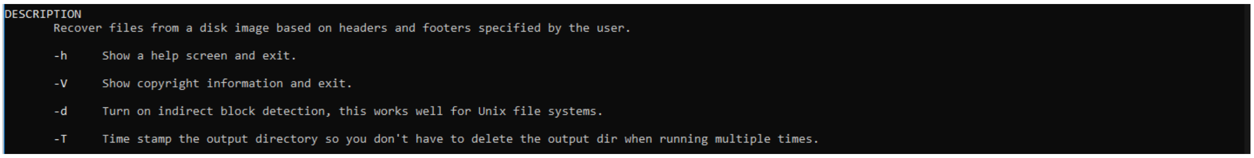


```
[liveuser@localhost-live ~]$ sudo yum install foremost
```

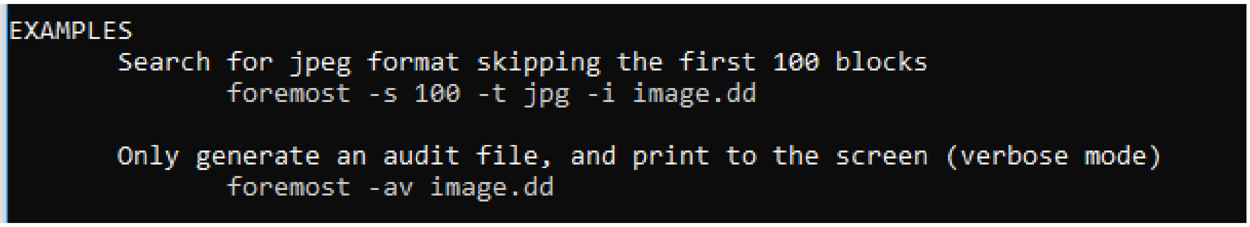
Next, we will look at the man page for Foremost [5]:



```
FOREMOST(8)                          System Manager's Manual                          FOREMOST(8)

NAME
       foremost - Recover files using their headers, footers, and data structures

SYNOPSIS
       foremost [-h] [-V] [-d] [-vqwQT] [-b <blocksize>] [-o <dir>] [-t <type>] [-s <num>] [-i <file>]

BUILTIN FORMATS
       Recover files from a disk image based on file types specified by the user using the -t switch.

       jpg    Support for the JFIF and Exif formats including implementations used in modern digital cameras.

       gif

       png

       bmp    Support for windows bmp format.

       avi
```



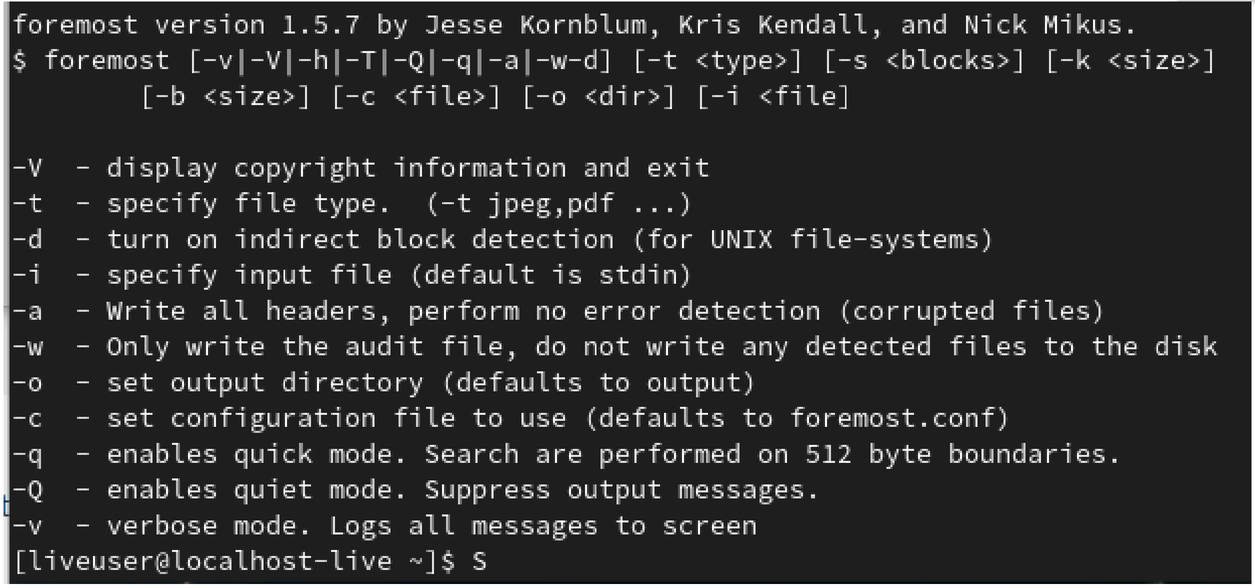
```
DESCRIPTION
       Recover files from a disk image based on headers and footers specified by the user.

       -h     Show a help screen and exit.

       -V     Show copyright information and exit.

       -d     Turn on indirect block detection, this works well for Unix file systems.

       -T     Time stamp the output directory so you don't have to delete the output dir when running multiple times.
```



```
EXAMPLES
       Search for jpeg format skipping the first 100 blocks
              foremost -s 100 -t jpg -i image.dd

       Only generate an audit file, and print to the screen (verbose mode)
              foremost -av image.dd
```

As we can see from the man page, it shows a quick description on how Foremost works and the various file types it supports. It also displays the various functions that it has and also the command line arguments that are built in. Furthermore, the last screenshot shows various examples of Foremost on various file types and disk drives [5].

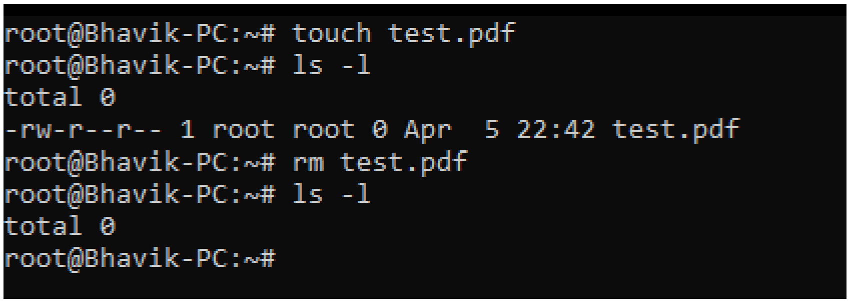
We can run the Foremost -h command which will output the various help commands from the program:



```
foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus.
$ foremost [-v|-V|-h|-T|-Q|-q|-a|-w-d] [-t <type>] [-s <blocks>] [-k <size>]
        [-b <size>] [-c <file>] [-o <dir>] [-i <file]

-V  - display copyright information and exit
-t  - specify file type.  (-t jpeg,pdf ...)
-d  - turn on indirect block detection (for UNIX file-systems)
-i  - specify input file (default is stdin)
-a  - Write all headers, perform no error detection (corrupted files)
-w  - Only write the audit file, do not write any detected files to the disk
-o  - set output directory (defaults to output)
-c  - set configuration file to use (defaults to foremost.conf)
-q  - enables quick mode. Search are performed on 512 byte boundaries.
-Q  - enables quiet mode. Suppress output messages.
-v  - verbose mode. Logs all messages to screen
[liveuser@localhost-live ~]$ S
```

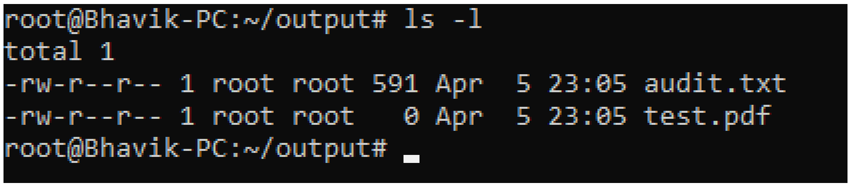
We can see the various authors of the program and the version that is installed. The in-line commands that we can run to recover various files from the disk is also stated.

Now, we will demonstrate an example that will utilize this program. We will create a pdf file, named “test.pdf” and delete it. We will then use Foremost to recover the deleted file.
Creating the file and deleting it:



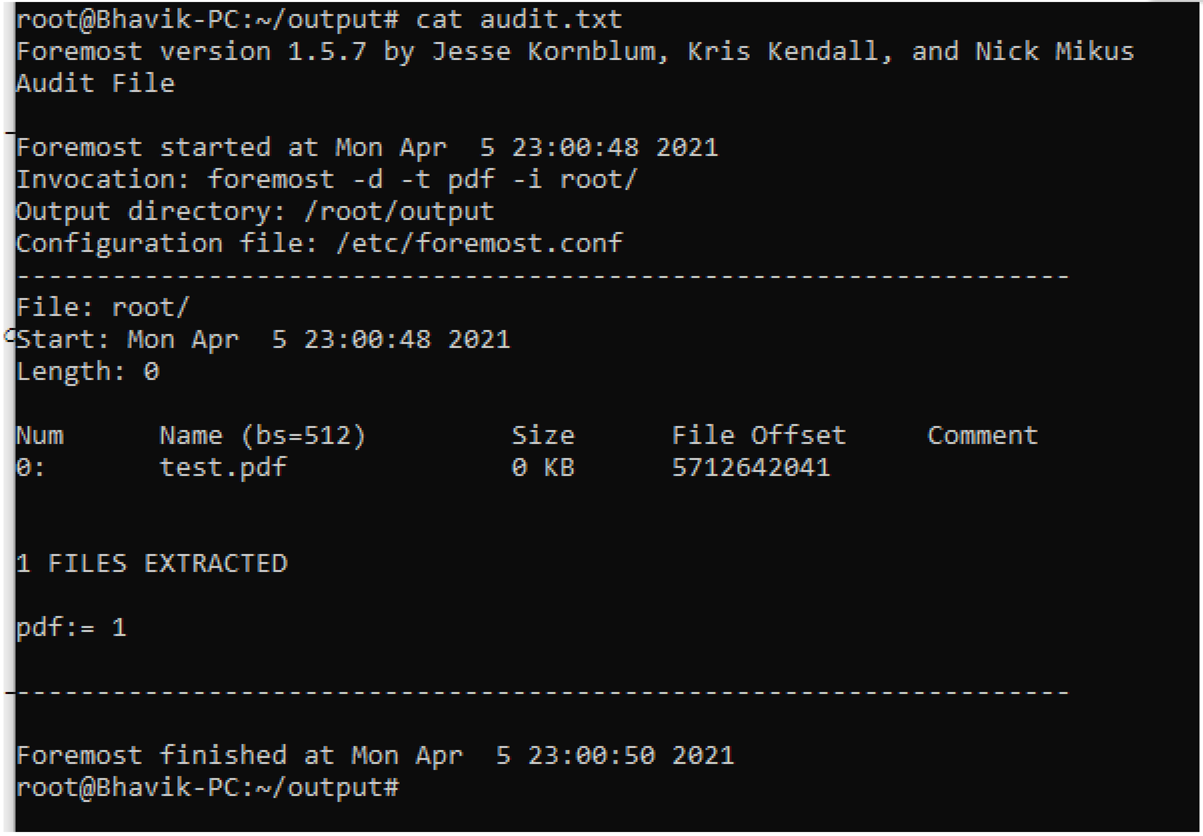
```
root@Bhavik-PC:~# touch test.pdf
root@Bhavik-PC:~# ls -l
total 0
-rw-r--r-- 1 root root 0 Apr  5 22:42 test.pdf
root@Bhavik-PC:~# rm test.pdf
root@Bhavik-PC:~# ls -l
total 0
root@Bhavik-PC:~#
```

We can run the “Foremost -d -t pdf -i root/” to allow Foremost to find the files that were deleted. After running that, we notice that there is “output” directory that was created. We can open that out and notice 2 files [4]:



```
root@Bhavik-PC:~/output# ls -l
total 1
-rw-r--r-- 1 root root 591 Apr  5 23:05 audit.txt
-rw-r--r-- 1 root root   0 Apr  5 23:05 test.pdf
root@Bhavik-PC:~/output# _
```

Using “cat” to display output.txt, we can see:



```
root@Bhavik-PC:~/output# cat audit.txt
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus
Audit File

Foremost started at Mon Apr  5 23:00:48 2021
Invocation: foremost -d -t pdf -i root/
Output directory: /root/output
Configuration file: /etc/foremost.conf
------------------------------------------------------------------
File: root/
Start: Mon Apr  5 23:00:48 2021
Length: 0

Num      Name (bs=512)         Size      File Offset     Comment
0:       test.pdf              0 KB      5712642041


1 FILES EXTRACTED

pdf:= 1

------------------------------------------------------------------

Foremost finished at Mon Apr  5 23:00:50 2021
root@Bhavik-PC:~/output#
```

This shows the Foremost was able to recover the PDF file that was deleted in the root folder. We are able to see the pdf in the output directory that Foremost created. The size of the file is correct as 0 as we did not populate the file. We just used “touch” to create the pdf file.

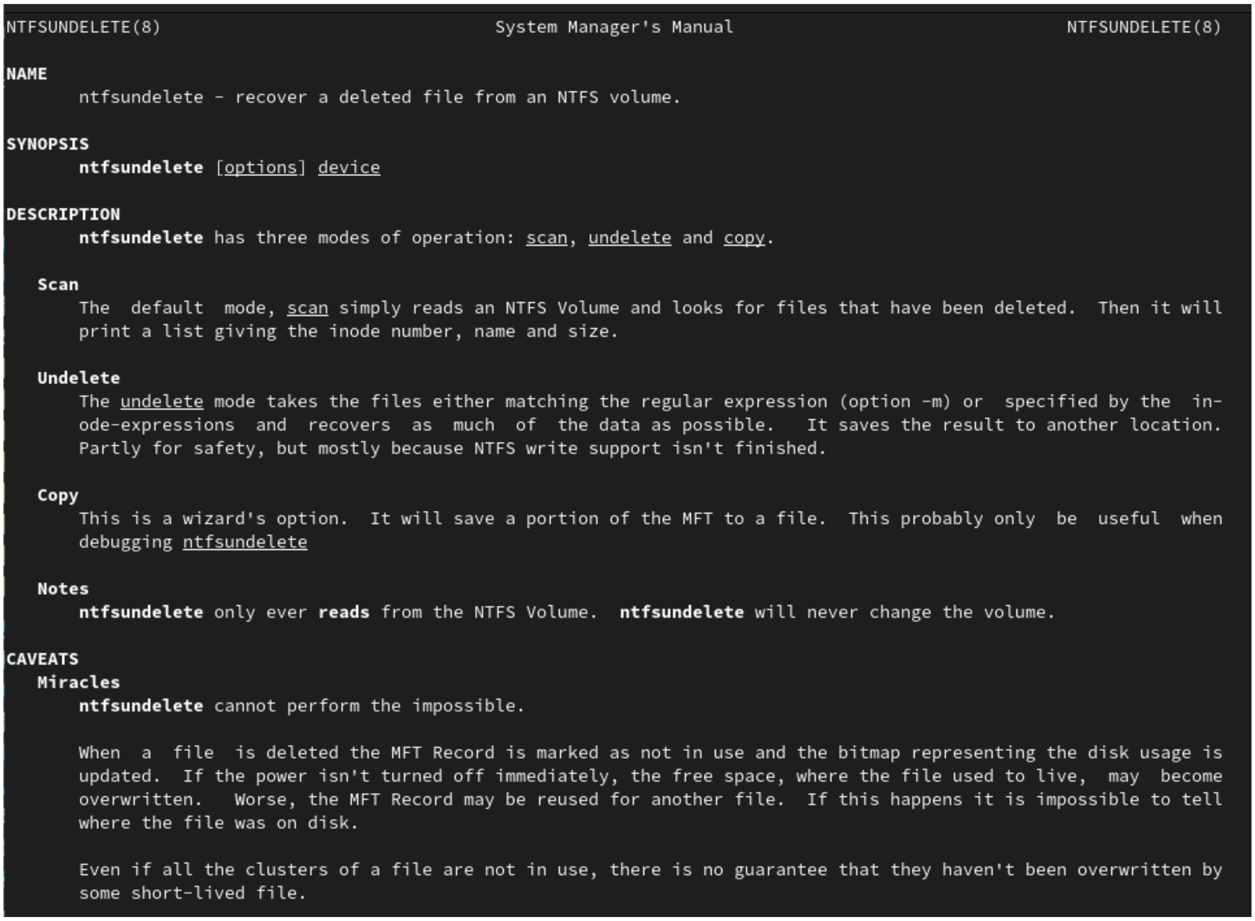
Therefore, we can see that Foremost works and gets the deleted file from the system that you specify. Foremost is an easy tool that you can run on the Linux command line and recover the deleted files directly from the live disk.

8 - ntfsundelete

The next recovery tool that we will demonstrate recovery with is ntfsundelete. It is a popular NTFS recoverer that will look solely on NTFS file systems and try to recover any files that have been deleted [7]. As the majority of the Windows devices and some Linux based operating systems are based on NTFS, it is important to have a reliable program that will quickly recover any lost files. Ntfsundelete is not restricted to the NTFS filesystem [7]. It can also recover data from FAT12/16/32 based file systems. It works on numerous file types and basically lists all of the various files that might be deleted from a disc image. We will see how that will look when showing examples of the program.

We do not have to worry about installation as many flavours of Linux have ntfsundelete already installed. But, if you do want to install it, you can run sudo yum install ntfs-3g. [7]

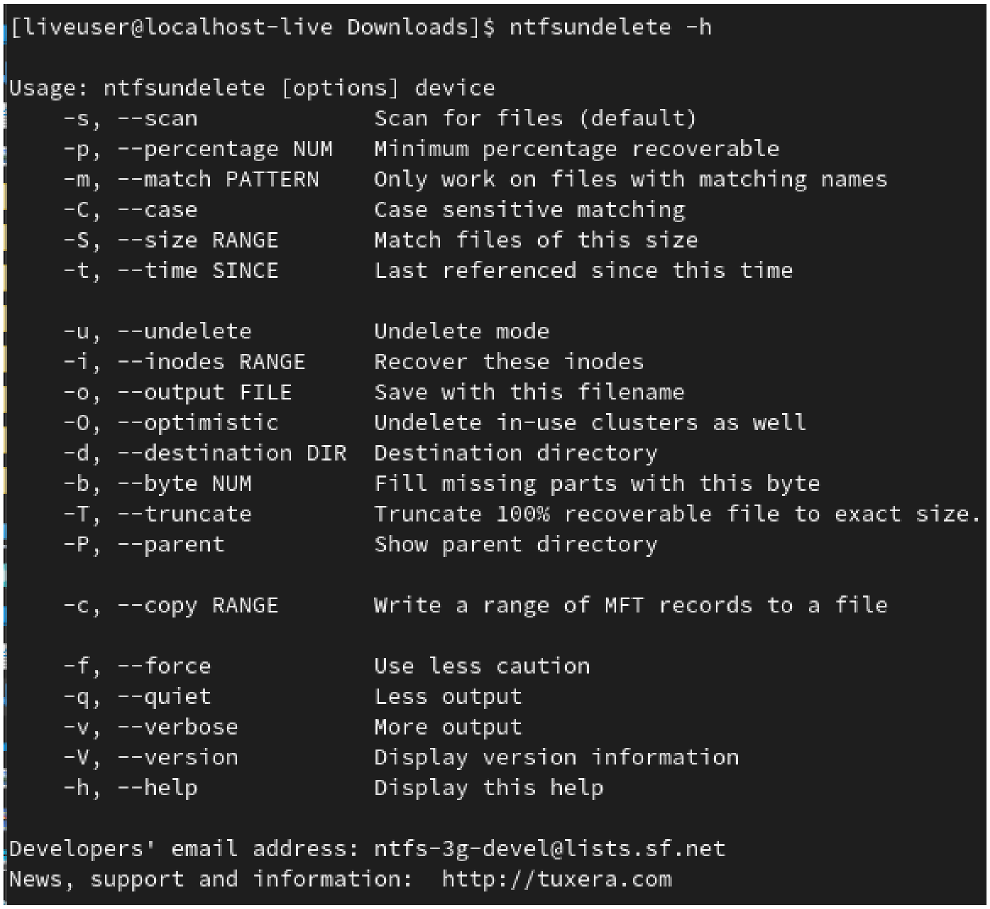
We can have a look at the man page for ntfsundelete [6]:



```
NTFSUNDELETE(8)                       System Manager's Manual                       NTFSUNDELETE(8)

NAME
       ntfsundelete - recover a deleted file from an NTFS volume.

SYNOPSIS
       ntfsundelete [options] device

DESCRIPTION
       ntfsundelete has three modes of operation: scan, undelete and copy.

   Scan
       The  default  mode, scan simply reads an NTFS Volume and looks for files that have been deleted.  Then it will
       print a list giving the inode number, name and size.

   Undelete
       The undelete mode takes the files either matching the regular expression (option -m) or  specified by the  in-
       ode-expressions  and  recovers  as  much  of  the data as possible.   It saves the result to another location.
       Partly for safety, but mostly because NTFS write support isn't finished.

   Copy
       This is a wizard's option.  It will save a portion of the MFT to a file.  This probably only  be  useful  when
       debugging ntfsundelete

   Notes
       ntfsundelete only ever reads from the NTFS Volume.  ntfsundelete will never change the volume.

CAVEATS
   Miracles
       ntfsundelete cannot perform the impossible.

       When  a  file  is deleted the MFT Record is marked as not in use and the bitmap representing the disk usage is
       updated.  If the power isn't turned off immediately, the free space, where the file used to live,  may  become
       overwritten.   Worse, the MFT Record may be reused for another file.  If this happens it is impossible to tell
       where the file was on disk.

       Even if all the clusters of a file are not in use, there is no guarantee that they haven't been overwritten by
       some short-lived file.
```

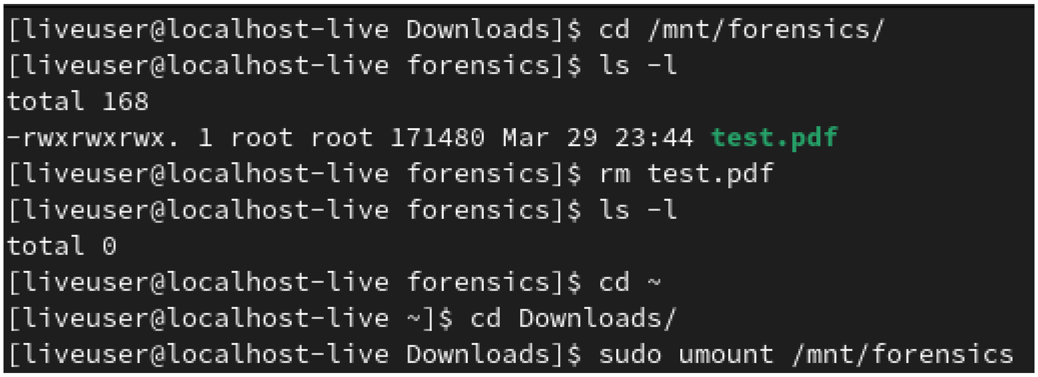
From this we can see the various functions, descriptions, and other warnings about how this program cannot perform miracles [6]. It also states that ntfsundelete has a scan, undelete, and copy functions that allow users to recover the files that they have deleted [6].

We can view the options that this program contains by running the ntfsundelete -h command:

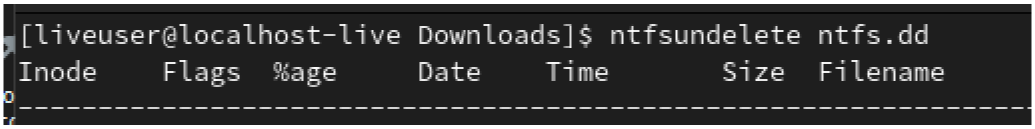


```
[liveuser@localhost-live Downloads]$ ntfsundelete -h

Usage: ntfsundelete [options] device
    -s, --scan              Scan for files (default)
    -p, --percentage NUM    Minimum percentage recoverable
    -m, --match PATTERN     Only work on files with matching names
    -C, --case              Case sensitive matching
    -S, --size RANGE        Match files of this size
    -t, --time SINCE        Last referenced since this time

    -u, --undelete          Undelete mode
    -i, --inodes RANGE      Recover these inodes
    -o, --output FILE       Save with this filename
    -O, --optimistic        Undelete in-use clusters as well
    -d, --destination DIR   Destination directory
    -b, --byte NUM          Fill missing parts with this byte
    -T, --truncate          Truncate 100% recoverable file to exact size.
    -P, --parent            Show parent directory

    -c, --copy RANGE        Write a range of MFT records to a file

    -f, --force             Use less caution
    -q, --quiet             Less output
    -v, --verbose           More output
    -V, --version           Display version information
    -h, --help              Display this help

Developers' email address: ntfs-3g-devel@lists.sf.net
News, support and information:  http://tuxera.com
```

Now, we will demonstrate an example that will utilize this program. First, we will mount the filesystem to the machine and then copy over a test pdf that will then be deleted. After unmounting, we will then run the ntfsundelete function to have a look at the table of deleted files. We then will undelete or copy over the deleted back into the file system.

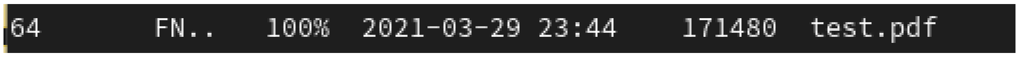


```
[liveuser@localhost-live Downloads]$ cd /mnt/forensics/
[liveuser@localhost-live forensics]$ ls -l
total 168
-rwxrwxrwx. 1 root root 171480 Mar 29 23:44 test.pdf
[liveuser@localhost-live forensics]$ rm test.pdf
[liveuser@localhost-live forensics]$ ls -l
total 0
[liveuser@localhost-live forensics]$ cd ~
[liveuser@localhost-live ~]$ cd Downloads/
[liveuser@localhost-live Downloads]$ sudo umount /mnt/forensics
```



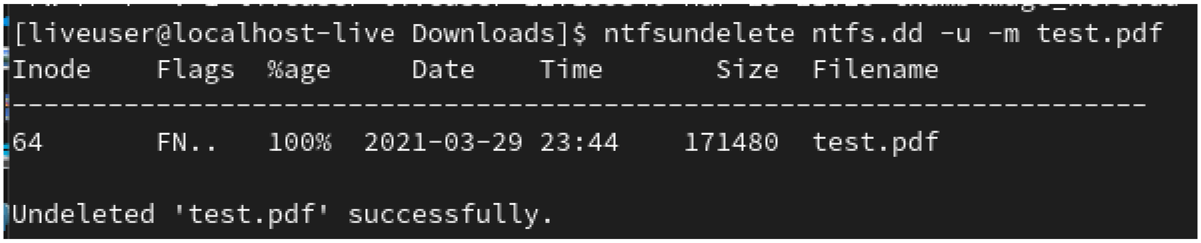
```
[liveuser@localhost-live Downloads]$ ntfsundelete ntfs.dd
Inode     Flags   %age      Date      Time        Size   Filename
-----------------------------------------------------------------------
```

Now we can see that in cluster 64, we have the test.pdf file. The 100% means that ntfsundelete can recover the entire file. 171480 is the size of the file that we are recovering and we can also see that time and data that the file was deleted.



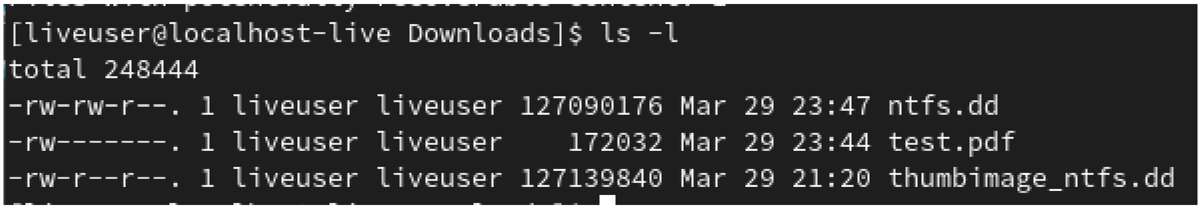
```
64        FN..    100%   2021-03-29 23:44    171480   test.pdf
```

To recover the file, we can run the command, ntfsundelete ntfs.dd -u -m test.pdf. After we run that command, we can see that the test.pdf file has been recovered in our Downloads folder.



```
[liveuser@localhost-live Downloads]$ ntfsundelete ntfs.dd -u -m test.pdf
Inode    Flags  %age     Date    Time       Size  Filename
-----------------------------------------------------------------------
64       FN..   100%  2021-03-29 23:44    171480  test.pdf

Undeleted 'test.pdf' successfully.
```

We can see that the file has been successfully recovered.



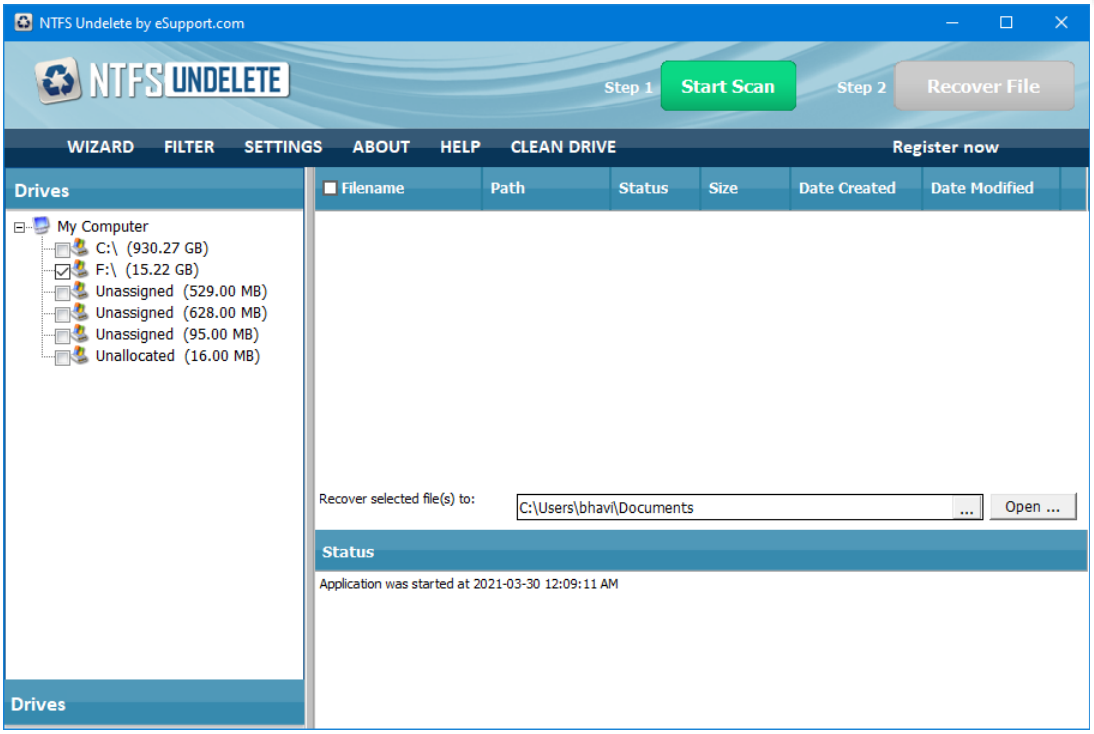
```
[liveuser@localhost-live Downloads]$ ls -l
total 248444
-rw-rw-r--. 1 liveuser liveuser 127090176 Mar 29 23:47 ntfs.dd
-rw-------. 1 liveuser liveuser    172032 Mar 29 23:44 test.pdf
-rw-r--r--. 1 liveuser liveuser 127139840 Mar 29 21:20 thumbimage_ntfs.dd
```

This also shows us that the test.pdf in its original size has been recovered successfully.

Ntfsundelete is an easy, quick tool that anybody can use to recover any deleted files from a NTFS or FAT file system. It can show you all the files that it can recover, with their sizes, dates, and what percentage of the file that program can recover. This allows the user to quickly see the details of the file and quickly recover it.

9 - Challenge with ntfsundelete

My group partner has given me a USB drive in which he deleted a file from. I will be using the ntfsundelete program that we will install on Windows. He told me that the file is an image file, and the name of the file is doge.png. We will be using the exe file that ntfsundelete offers for the Windows operating system as we have already demonstrated how it works on Linux before using the command line.
When plugging in the USB and opening up ntfsundelete, we can see that it displays 2 disks, the C drive and the F drive. For our purposes, the F drive contains the deleted file.



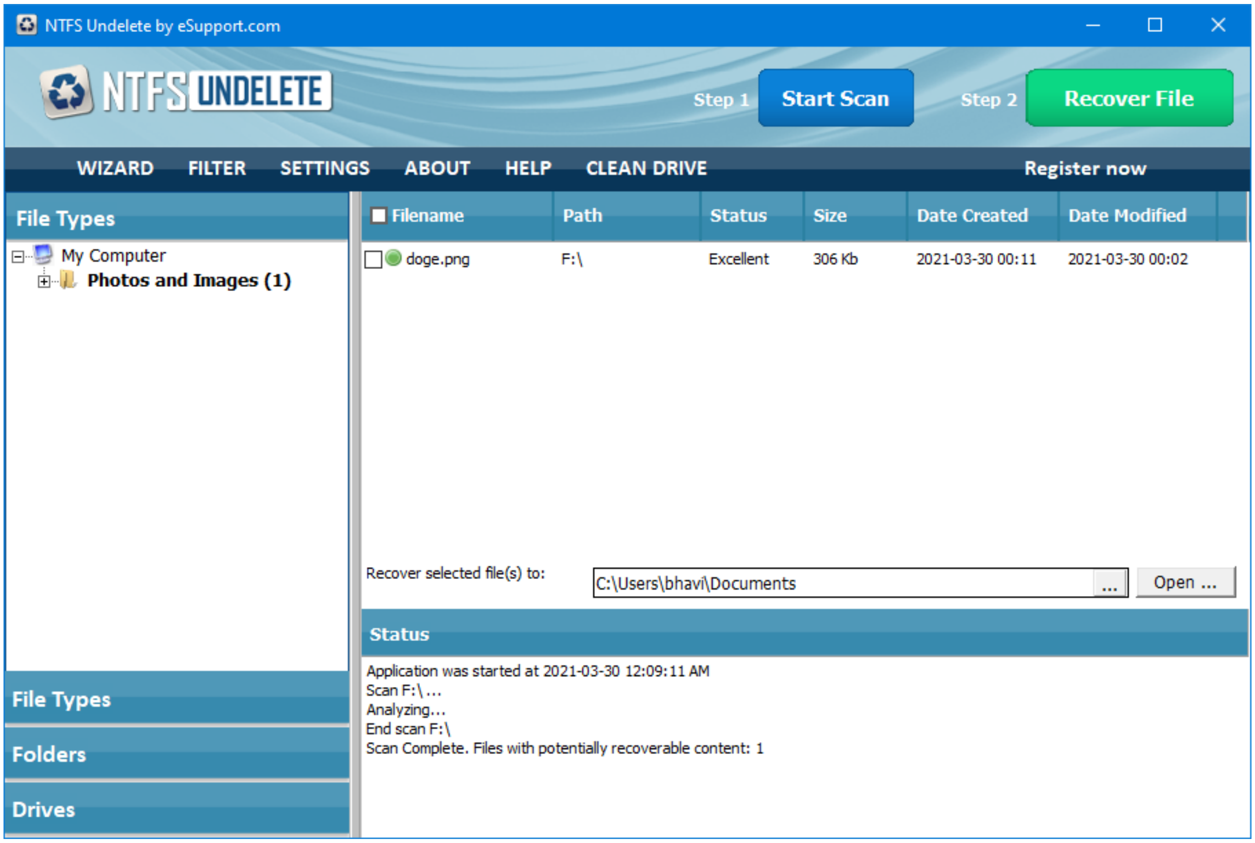
- My Computer
  - □ C:\ (930.27 GB)
  - ☑ F:\ (15.22 GB)
  - □ Unassigned (529.00 MB)
  - □ Unassigned (628.00 MB)
  - □ Unassigned (95.00 MB)
  - □ Unallocated (16.00 MB)

□ Filename	Path	Status	Size	Date Created	Date Modified

Recover selected file(s) to: C:\Users\bhavi\Documents ... Open ...

```
Application was started at 2021-03-30 12:09:11 AM
```

We will now “Start scan” on the F drive to let the program scan for any deleted files. After running the scan on the F drive, it was quick to find the deleted media file, doge.png.



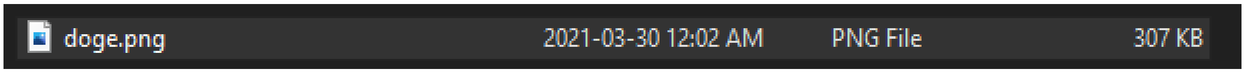
- My Computer
  - **Photos and Images (1)**

□ Filename	Path	Status	Size	Date Created	Date Modified
□ ● doge.png	F:\	Excellent	306 Kb	2021-03-30 00:11	2021-03-30 00:02

Recover selected file(s) to: C:\Users\bhavi\Documents ... Open ...

```
Application was started at 2021-03-30 12:09:11 AM
Scan F:\ ...
Analyzing...
End scan F:\
Scan Complete. Files with potentially recoverable content: 1
```

We can see that the status of the file is excellent, which means that program will have no trouble recovering it and we can see the size of it too. We will now click, “Recover file” and the program should be able to recover it.



doge.png 2021-03-30 12:02 AM PNG File 307 KB

We can see that the file has been recovered and we can see the content of it. This shows that the program was able to successfully recover the file and not lose any data from it.

We can have demonstrated that by using ntfsundelete, we can easily recover the file that was located on the USB flash disk and see the deleted file.

Ntfsundelete is a quick and easy tool to recover lost data from any system running NTFS.

10 - References

[1] CGSecurity, “Testdisk,” 27-Oct-2019. [Online]. Available: https://www.cgsecurity.org/wiki/TestDisk.[Accessed: 06-Apr-2021].

[2] C. Grenier, *TestDisk Documentation*, 04-Oct-2020. [Online]. Available: https://www.cgsecurity.org/testdisk.pdf. [Accessed: 05-Apr-2021].

[3] “The Sleuth Kit - analyze disk images and recover files,” *LinuxLinks*, 06-Jul-2019. [Online]. Available: https://www.linuxlinks.com/thesleuthkit/. [Accessed: 06-Apr-2021].

[4] E. Docile, “How to recover deleted files with foremost on Linux,” Linux Tutorials - Learn Linux Configuration, 28-May-2020. [Online]. Available: https://linuxconfig.org/how-to-recover-deleted-files-with-foremost-on-linux. [Accessed: 07-Apr-2021].

[5] foremost(1) - Linux man page. [Online]. Available: https://linux.die.net/man/1/foremost. [Accessed: 07-Apr-2021].

[6] ntfsundelete(8) - Linux man page. [Online]. Available: https://linux.die.net/man/8/ntfsundelete. [Accessed: 07-Apr-2021].

[7] S. Shovon, “Recover Removed Files from Windows NTFS Drive using Live Ubuntu DVD,” Linux Hint, 02-Apr-2019. [Online]. Available: https://linuxhint.com/recover_files_ntfs_live_ubuntu/. [Accessed: 07-Apr-2021].